Mestrado Integrado em Engenharia Química

Proper Orthogonal Decomposition of Turbulent Flows

Tese de Mestrado

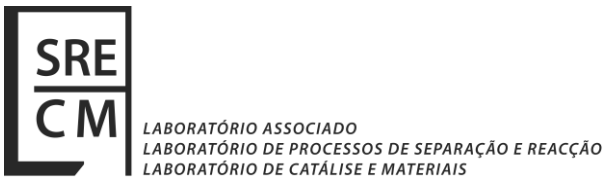
de

José Pedro Nunes da Silva Torres

Desenvolvida no âmbito da unidade curricular de Dissertação

realizado em

LSRE – Laboratory of Separation and Reaction Engineering



Orientadores: **Doutor Ricardo Jorge Santos**

Doutor Nelson Daniel Gonçalves

Departamento de Engenharia Química

Julho de 2017

“Up ahead they's a thousan' lives we might live,

but when it comes

it'll on'y be one.”

— John Steinbeck, The Grapes of Wrath

Agradecimentos

Estranhamente, esta página é a que me custa mais a escrever durante toda a tese. Custa porque nunca conseguirei agradecer o suficiente às pessoas que me ajudaram a fazer este trabalho.

Aos meus orientadores. O professor Ricardo que viu em mim alguém capaz de fazer parte do MixingGroup, e que me foi ensinando coisas que nunca pensei vir a compreender. Fez-me ver que, nas palavras dele, o que é fácil de compreender é aborrecido. É a tentar aprender o difícil que o trabalho se torna interessante.

Ao Nelson Gonçalves, que se mostrou sempre disponível para me ajudar em tudo que precisava, e que me lembrou do quanto eu gostava de matemática quando estava no secundário.

Não posso deixar de agradecer à professora Madalena Dias e ao professor José Carlos Lopes por me receberem no LSRE-LCM e no MixingGroup, por acreditarem nas minhas competências dentro de um grupo com expectativas elevadas.

Aos meus amigos. Tanto aos que ficaram nas Almoçaradas na faculdade, como os que foram escrever as teses noutro lado. Os últimos anos convosco foram fantásticos. Se pudesse, repetia-os até ao fim da minha vida. Que nunca deixem de se rir comigo (nem de mim).

À minha família. Não seria quem sou se não fosse por eles e por causa deles. A minha Mãe, a mulher mais lutadora que conheço, que combateu sempre contra tudo e todos para garantir um futuro melhor para mim e para a minha irmã. Ao meu Pai, que cujo sentimentalismo passou completamente para mim, que fez sempre tudo a tentar ser a melhor pessoa que pudesse. À minha Irmã, que espero ter sido tanto um exemplo para ela como ela tem sido para mim. E à minha Avó, que tantas vezes coseu o meu traje, sem nunca se queixar de tantos rasgões que ele tinha.

Finalmente, à rapariga que conheci antes de entrar na faculdade. Faltam-me as palavras para te agradecer por tudo, pelo exemplo que és e pela força que me dás. Pode ser que no resto da nossa vida as arranje para te as dizer. Por enquanto fica a mesma palavra de sempre, cada vez com mais significado: amo-te.

Este trabalho foi em parte financiado pelos projetos 2Dmix e POCI-01-0145-FEDER-016851 - Laboratório Associado LSRE-LCM - financiado pelo Fundo Europeu de Desenvolvimento Regional (FEDER), através do COMPETE2020 – Programa Operacional Competitividade e Internacionalização (POCI) e por fundos nacionais através da Fundação para a Ciência e a Tecnologia. Special Thanks to Prof. Alain Liné for support on the POD code.

Resumo

O fenómeno da turbulência, enquanto um processo aleatório, torna necessário o estudo de fluxos turbulentos recorrendo ao uso de ferramentas estatísticas. Este trabalho diz respeito à aplicação dessas ferramentas em dois tipos de misturadores: o misturador de jatos opostos confinados (CIJ) e o misturador NETmix. Há uma consideração especial sobre a aplicação da decomposição ortogonal própria (POD), como uma operação estatística que resume as principais características de um fluxo turbulento em poucas componentes.

Os tipos de análise utilizados para ambos os reatores são: histogramas e funções de densidade de probabilidade, correlação de dois pontos e cálculo de escalas integrais de turbulência, espectros de turbulência (apenas para o NETmix) e o POD já mencionado. Para obter todos os dados necessários para aplicar essas ferramentas, várias simulações de Dinâmica de Fluidos Computacionais (CFD) em duas dimensões foram feitas para cada misturador, a vários números de Reynolds.

Os resultados da análise dos CIJ foram comparados com trabalhos anteriores. Para o NETmix, foi feita uma abordagem usando condições de fronteira periódicas para reproduzir a repetição da rede de mistura, levando a uma análise suplementar sobre como o número de câmaras do modelo em CFD afeta a distribuição dos momentos estatísticos da velocidade. Isso permitiu uma decisão sobre a câmara mais adequada para realizar as outras análises.

O POD permitiu uma reconstrução do campo de velocidade de ambos os misturadores, recorrendo a poucas componentes, a partir das quais as principais estruturas eram identificáveis.

Palavras Chave (Tema): Turbulência, CIJ, NETmix, CFD, histograma, função de distribuição de probabilidade, escalas integrais de turbulência, espectro de turbulência, decomposição ortogonal própria.

Abstract

The phenomenon of turbulence as a random process makes it necessary to study turbulent flows with the use of statistic tools. This work concerns the application of these tools to two types of mixers: the Confined Impinging Jets mixer (CIJ) and the NETmix mixer. There is a special regard on the application of Proper Orthogonal Decomposition (POD), as a statistical operation that summarizes the main characteristics of a turbulent flow in few components.

The types of analysis used for both the reactors are: histograms and probability density functions, two-point correlation and calculation of turbulence integral lengthscales, turbulence spectra (only for the NETmix) and the already mentioned POD. To get all the data necessary to apply these tools, various 2D Computational Fluid Dynamics (CFD) simulations were done for each mixer, with various Reynolds numbers.

The results of the analysis of the CIJs were compared with previous work. For the NETmix, an approach using periodic boundary conditions to model the repeating structure of the mixer was done, leading to an extra analysis on how the number of chambers of the CFD model affects the distribution of velocity statistics. This allowed for a decision on the most adequate chamber on which to realize the other analysis.

The POD enabled a reconstruction of the velocity field of both mixers, resorting to few components, on which the major structures were identifiable.

Keywords (Theme): Turbulence, CIJ, NETmix, CFD, histogram, probability distribution function, turbulence integral lengthscales, turbulence spectrum, proper orthogonal decomposition.

Declaração

Declaro, sob compromisso de honra, que este trabalho é original e que todas as contribuições não originais foram devidamente referenciadas com identificação da fonte.

José Pedro Nunes da Silva Torres

Julho de 2017

Contents

List of Figures

List of Tables

Nomenclature

Upper-case Roman

C	Kolmogorov Constant	
D	Diameter of the chamber	mm
E	Velocity spectrum	
$\boldsymbol{I}$	Identity matrix	
$\mathcal{I}$	Turbulence Intensity	
K	Snapshot *kernel*	
L	Length of the chamber	mm
L_{11}	Turbulence Integral Scale	mm
L	Turbulence Integral Scale	mm
$\mathcal{L}$	Lagrangian multiplier	
M	Number of time-steps (snapshots)	
N	Number of events	
$\mathcal{N}$	Number of modes	
P	Probability density function	
R	Two-point correlation	
$\boldsymbol{R_X}$	Auto-covariance tensor of $\boldsymbol{X}$	
Re	Reynolds number	
T	Time period	s
$\boldsymbol{U}$	Velocity field	
$\widetilde{\boldsymbol{U}}_k$	Instantaneous reconstructed velocity field	
W	Hammond window function	
$\boldsymbol{X}$	Set of vectors	
$\boldsymbol{Y}$	Set of Proper Orthogonal Decomposition modes	

Lower-case Roman

a	Proper Orthogonal Decomposition coefficient	
c	Range in a histogram	
d	Diameter of the inlets	mm
d_h	Hydraulic diameter	mm
f	Frequency	s^{-1}
l	Channel length	mm
p	Pressure	Pa
p_0	Atmospheric pressure	Pa
r	Displacement between two points	mm
t	Time	s
u	Velocity	m/s
$\boldsymbol{u}$	Velocity vector	

$\boldsymbol{v}'$	Velocity fluctuation around u_{inj} normalized with u_{inj}	
x	Position	mm
$\boldsymbol{x}$	Position vector	
x axis	Cartesian horizontal coordinate	
w	Depth of a channel	mm
y axis	Cartesian vertical coordinate	

Lower-case Greek

$\boldsymbol{\alpha}_i$	Eigenvector	
β	Hammond window function parameter	
ε	Rate of dissipation of turbulent kinetic energy	
λ	Wavenumber	m^{-1}
λ_i	Eigenvalue of eigenvector i	
μ	viscosity	$\mathrm{Pa \cdot s}$
ξ	Hammond window function parameter	
ρ	Density	$\mathrm{kg/m^3}$
σ	Standard deviation	
τ_x	x component of the shear stress	Pa
υ	Random event	

Abbreviations

2D	Two-dimensional
3D	Three-dimensional
CIJ	Confined Impinging Jets (mixer)
CFD	Computational Fluid Dynamics
FEUP	Faculdade de Engenharia da Universidade do Porto
LSRE	Laboratory of Separation and Reaction Engineering
MB	Megabytes
NUB	NETmix Unit Block
PDF	Probability Density Function
POD	Proper Orthogonal Decomposition
RAM	Random Access Memory
RIM	Reaction Injection Molding

1 Introduction

1.1 Motivation and Relevance

In opposition to three-dimensional turbulent flows, where the energy of the large scales is transferred to smaller scales [1], in two-dimensional turbulence the small scales feed larger eddies, a phenomenon known as the inverse energy cascade [2]. In addition to this, as 3D turbulence enables stretching of vortices and 2D constrains the movement in the third dimension, there is a conservation of enstrophy and two energy cascades: the mentioned inverse energy cascade and the enstrophy cascade, where the energy is transferred to smaller lengthscales [3, 4].

Turbulence is especially important when it concerns mixing. A mixing device that can explore the advantages of turbulence is the confined impinging jets (CIJ) mixer, which principle of function is based on the notion that the energy dissipated from the two jets, after impingement in the confined space, allows for mixing to be achieved rapidly and without movable mechanical parts, such as stirring devices [5].

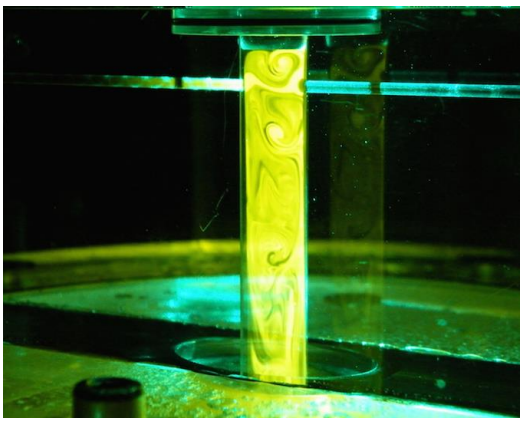


Figure 1 CIJ Mixer

It is important to note that CIJs have two main flow regimes: one at a low Reynolds number, where the flow is steady and the fluids injected from opposed jets are kept almost segregated; and a chaotic or dynamic flow field where a street of vortices is formed upon the impingement of the opposed jets [6]. These vortices are considered coherent structures [7], which are flow formations in turbulence that are coherent and easily recognizable [8].

The flow in this type of mixers is mostly affected by the dynamics of the plane of the jets, defined by the jets axis and chamber axis [3], which has led to extensive studies of CIJs flow dynamics in 2D physical models, validated and compared with actual 3D physics by Santos et al

[9, 10]. Throughout this thesis, a more detailed explanation of the behavior of CIJs will be given.

Another mixer that uses this hydrodynamic behavior is the NETmix, which exhibits the properties of impinging jets in a network of small channels and chambers. This static mixer was introduced first by Laranjeira [11], but a short explanation will also be given ahead in this thesis.

Proper Orthogonal Decomposition (POD) is a powerful method for analyzing data that focuses on using small dimensional approaches to high dimensional processes [12]. One of the first explorations of this tool in flow dynamics was made by Lumley in 1967, based on the idea that spatial velocity correlations can be decomposed orthogonally as a method of identifying turbulent coherent structures [13], such as the vortices created by the CIJs.

To study the application of POD in these mixers, flows at different Reynolds numbers are simulated using CFD, and the simulation results are treated with MATLAB, which code used is also subject to alterations in order to optimize sampling and execution time.

1.2 Thesis Objectives and Layout

This thesis has the objective of exploring two different types of mixers, the CIJ and the NETmix, via a series of statistical tools, one of which is the Proper Orthogonal Decomposition, that offers a new perspective on the existence of coherent structures and their associated kinetic energy in the flow of each mixer.

The layout is as follows.

In Chapter 2, a brief introduction to how statistics are used in turbulence is summarized in Section 2.1, as well as some tools that are used in particular, which are explained in the following subsections: Histograms and the Probability Density Function (Subsection 2.1.1), Two-Point Correlation and Turbulence Integral Lengthscales (Subsection 2.1.2), Turbulence Spectra (Subsection 2.1.3) and finally Proper Orthogonal Decomposition (Subsection 2.1.4). In Section 2.2, a short introduction to the studies done on both the mixers explored in this work is done.

In Chapter 3, the dynamics of the turbulent flow on the CIJs are the main focus, starting with an explanation of the model used in the simulations (in Section 3.1), followed by the methodology and presentation of results of several statistics: Probability Density Functions (Section 3.2), Turbulence Integral Scales (Section 3.3) and POD (Section 3.5). In Section 3.4, the Turbulence Spectra is mentioned, and a work on this tool applied to the CIJs is referenced.

Chapter 4 consists of a similar analysis as Chapter 3, applied to the NETmix Unit Block, with an explanation of the model used in the CFD calculations in Section 4.1, followed by the methodology and presentation of results of the same tools as before: Probability Density

Functions (Section 4.2), Turbulence Integral Scales (Section 4.3), Turbulence Spectra (Section 4.4) and POD (Section 4.5).

In Chapter 5, some conclusion of the work done are exposed, as well as some limitations of the work and the proposal of future work.

2 State of the Art

2.1 Statistical Description of Turbulence

One of the symptoms of the turbulence syndrome, as coined by R. W. Stewart, is that the velocity field is such a complicated function of space and time that a statistical description is easier than a detailed description [14]. Because of this, turbulence analysis is mainly focused on numerical manipulation, which is compatible with computational fluid dynamics (CFD) simulations, from which one can retrieve the numerical data necessary to study this type of flow. This analysis is achieved resorting to a vast set of tools, two of which are the probability density functions (PDFs) and Turbulence spectra.

One particular aspect of turbulence analysis is how most of the statistical analyses of turbulent flows are based on the idea of an *ensemble average* in one form or another [15]. This concept of ensemble average consists on the extrapolation of an arithmetic average of random events to an infinite number of experiments. So, to put it in equation form, the ensemble average of the random event v, is

$$\langle v \rangle = \lim_{N \to \infty} \frac{1}{N} \sum_{n=1}^{N} v_n \tag{2.1}$$

where N the number of events is infinite. Of course, this is impossible to determine experimentally. However, what one can do is use the arithmetic average of a N large enough so that this average can be an *estimator* of the ensemble average. In other words, if we take enough realizations of a random event and average them, we can claim that we have a close approximation of that event's ensemble average.

Another important term associated with turbulence is the meaning of *fluctuations*. These are differences from the mean that occur in turbulence, where energy is transferred to and from turbulent eddies making some properties fluctuate away from the ensemble average. Resorting to the random event v, turbulent fluctuations are calculated as

$$v' = v - \langle v \rangle \tag{2.2}$$

It is easy to conclude that the ensemble average of the fluctuation is 0. However, if one calculates the ensemble average of the squared fluctuations, one gets a measure that is known in statistics as *variance*.

In this field of study, these terms follow a naming convention where variance is named the *second central moment of* v, and the ensemble average is the *first moment*. This naming convention is associated with the application of equation (2.1), according to the power of v_n.

In equation form, the m-th moment of the random event is defined as:

$$\langle v^m \rangle = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} v_n^m \tag{2.3}$$

Variance is considered the second central moment because it is defined by the difference of the second moment and the first moment squared, which is the square of the ensemble average [15].

These terms are of the upmost importance in the following sections, where the statistical tools will be described more thoroughly.

2.1.1 Histograms and the Probability Density Function

A lot of statistical analysis focuses on *random* events and statistical calculations regarding them. This is due to the fact that the velocity field is random in turbulent flows. It is a common mistake to incorrectly add significance to the meaning of *random*, arguing against the fact that turbulence is a random phenomenon. However, randomness only means that an event is not certain nor impossible. Regarding the velocity field in turbulent flows, its value is not the same every time, even under the same conditions. This means that in this type of flows, it is far easier to determine the probability of an event than to try to predict the value of such event [1].

To analyze the probability of these random events, it is necessary to determine experimentally the frequency of occurrence of a given value in N realizations. To do this, one must first take the overall range of the measured variable and divide said range in several bins, or subranges, where each value allocation is counted and then, after all the realizations are allocated, this count is divided by the number of realizations, N. This produces a *histogram*, where the sum of count value of each bin (divided by the number of realizations) is equal to 1. This is analogous to the probability of each realization of the random event resulting in a value that belong in each bin. The higher *N*, the closer the probability to the actual expected value, according to the law of large numbers.

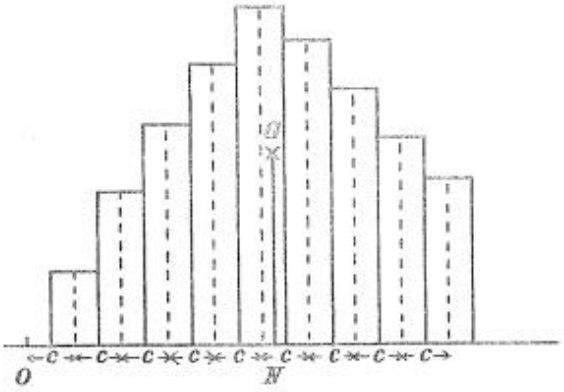


Figure 2 First introduction of the histogram [16]

If the bin size tends to zero, dividing the histogram by the bin size will create a limiting curve called the *probability density function*. Thus, the PDF is the probability *per unit distance* in the sample space, and it must be stressed that it completely characterizes the random event in study [1].

When taking into consideration physical data, the amount of realizations is defined by the observed time period, T. So the probability density function of $v(t)$ can be defined as the ratio of the time that $v(t)$ falls inside the defined range $(v, v + c)$, T_v, and the total time, T, for a small c [17]. So, the probability density function, $P(v)$, can be defined as follows.

$$P(v) = \lim_{c \to 0} \frac{1}{c} \left(\lim_{T \to \infty} \frac{T_v}{T} \right) \tag{2.4}$$

In turbulence, it is important to refer to the Normal distribution (also known as Gaussian distribution), defined by

$$P(v) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(v - \langle v \rangle)^2}{2\sigma^2}} \tag{2.5}$$

In which $\langle v \rangle$ is the mean (the ensemble average explained before) and σ is the *standard deviation*, which is the square root of the variance. As it is clear to see, this distribution is only defined by two variables that are easy to calculate from the data gathered in the realizations of an experiment. We will see later that turbulence is not completely described by these distributions, yet this is a simple and strong approach to many turbulent flows.

2.1.2 Two-Point Correlation and Turbulence Integral Lengthscales

The autocovariance of two points at the same time is the simplest statistic of a random field that contains information on the spatial structure, and is often known as *two-point correlation*. This correlation, in turbulence, indicates the relations between neighboring velocity fluctuations [18]. It can be written as

$$R(r, x, t) = \langle v(x, t) v(x + r, t) \rangle \tag{2.6}$$

where x is the position, r is the displacement between the two points, v is the random variable (velocity) and t is the time.

From this, it is possible to define the *turbulence integral lengthscale* of the flow. This scale represents the distance over which the fluctuating velocity field is correlated [1]. This distance is the size of the largest turbulent eddies [19]. The integral lengthscale, L_{11}, is taken from the integral of the two-point correlation, from the maximum point to the point where the correlation is zero, as shown in the following equation

$$L_{11}(x) = \frac{1}{T}\int_0^t \int_0^{R=0} \frac{R(r,x,t)}{R(0,x,t)}\, dr\, dt \tag{2.7}$$

As depicted in the equation, this value is divided by $R(0,x,t)$, since the correlation between a point and itself is maximum, normalizing the scale to a maximum of 1.

2.1.3 Turbulence Spectra

The information contained in the two-point correlation can be also expressed in a *wavenumber spectrum*, since in homogenous turbulence R is independent of x [1]. The velocity spectrum of a turbulent flow is calculated from the two dimensional Fourier transform of its velocity autocorrelation,

$$E(f) = \left[\int_{-\infty}^{\infty} e^{-2\pi i f t} u(t) dt\right]^2 \tag{2.8}$$

Normalized in this work by the injection velocity, u_{inj}, like so:

$$\lambda = \frac{f}{u_{inj}} \tag{2.9}$$

The velocity spectrum is E, and it is the contribution of the covariance of the velocity with frequency f, normalized by the injection velocity to get the wavenumber λ.

As referenced by the already mentioned work of Gonçalves [3], the enstrophy and inverse energy cascade is tightly linked with the power spectra, as noted by the work of Amarouchene and Kellay [20]. This is related to the various ranges existent in a turbulent flow: the energy containing range, the inertial range, and the dissipation range. The energy containing range is at the largest wavenumbers, the dissipation range for the smallest and the inertial range is in between them.

Kolmogorov's third hypothesis, known as the second similarity hypothesis, is that for turbulent flows with high enough Reynolds numbers, the statistics of motions with a scale between the largest eddies and the smallest, dissipative eddies, are described by a universal form determined only by the rate of dissipation of turbulent energy, independent of viscosity [1].

This range is the inertial range, and, according to this hypothesis, its spectrum is described by the equation

$$E(\lambda) = C\varepsilon^{\frac{2}{3}}\lambda^{-\frac{5}{3}} \tag{2.10}$$

where C is a universal constant (Kolmogorov's constant) and ε is the dissipation rate. So, in a power spectrum of a turbulent flow, the inertial range has a slope of $-5/3$, and a change of slope indicates a change of range. Figure 3 shows an example of what this change of slope looks like in a power spectra.

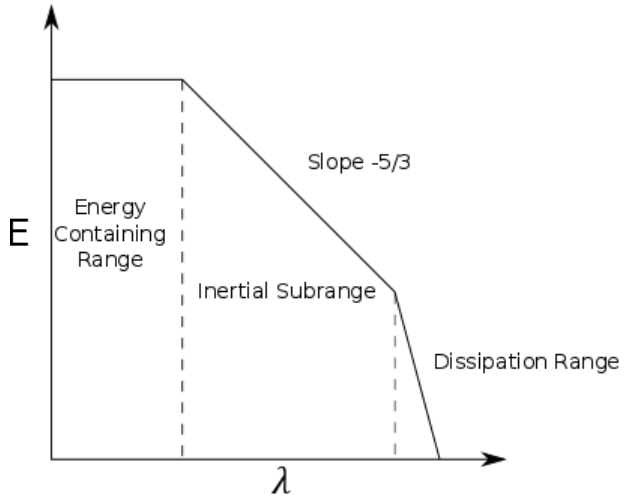


Figure 3 Power spectrum example

The change into the dissipation range is the injection of turbulence into the direct cascade (dissipating the energy), according to Boffetta and Ecke [2].

2.1.4 Proper Orthogonal Decomposition

Proper orthogonal decomposition provides the most efficient way of capturing the dominant components of an infinite-dimensional process with only finitely many, and often surprisingly few, "modes" [21].

As it was said in the introduction, one of the first uses of POD in turbulence was by Lumley in 1967, but according to Holmes [21], other uses of either POD or similar methods have been made earlier in other fields of work by Kosambi [22], Loève [23] and Karhunen [24]. These are the publications from which the Kosambi–Karhunen–Loève theorem, as referenced in [25], or simply Karhunen-Loève theorem, arose as way of applying the POD to infinite-dimensional spaces, as it uses single parameter functions instead of vectors [12].

This procedure is used, in the context of turbulence, to extract dominant features and trends, coherent structures, that exhibit patterns in space and also in time [21]. This is particularly relevant in mixers because it is in these structures that relevant degrees of mixing occur.

There is more than one method and interpretation for POD, as explored by Liang [12], but all of these methods follow similar steps that were used in this work, and will be explained next.

An approach given by Liné [26] used POD in regard to the information theory. This theory says the amount of information contained in a probability distribution is named "information entropy". It is established that maximum entropy means that the probability distribution associated is the most likely to occur, i.e., the elements with the maximum entropy are the ones that contain the most information.

Establishing a connection to turbulent flows, POD defines each mode as an equivalent to what a probability function is in the context of the information theory. This means that the first mode, or the most important, is the one more likely to occur in the flow.

Mathematically, we can state that if $\boldsymbol{X}$ is a set of vectors and $\boldsymbol{Y}$ is a set of modes,

$$\boldsymbol{Y}_1 = \sum_{i=1}^{n} \boldsymbol{\alpha}_{i,1} \boldsymbol{X}_i = \boldsymbol{\alpha}_1^T \boldsymbol{X} \tag{2.11}$$

This means that $\boldsymbol{Y}_1$ is the first principal component, or mode, where $\boldsymbol{\alpha}_1$ is a constant vector.

Sirovich used this notion together with the *snapshot method* [27], which claims that for a M number of time-steps (snapshots), which complete enough time so that the values of a velocity field for different time-steps are uncorrelated, we have a *kernel* $\boldsymbol{K}$ defined by:

$$\boldsymbol{K} = \frac{1}{M} \sum_{i=1}^{n} \boldsymbol{X}_i \boldsymbol{X}_i^T \tag{2.12}$$

This is essentially the auto-covariance tensor $\boldsymbol{R}$ of $\boldsymbol{X}$:

$$\boldsymbol{R}_{\boldsymbol{X}} = \frac{1}{M} \boldsymbol{X}\boldsymbol{X}^T = \begin{bmatrix} \overline{\boldsymbol{X}^2{}_1} & & \\ & \ddots & \\ & & \overline{\boldsymbol{X}^2{}_n} \end{bmatrix} \tag{2.13}$$

Combining Equations (2.11) and (2.13), we get

$$\boldsymbol{R}_{\boldsymbol{Y}_1} = \frac{1}{M} \boldsymbol{\alpha}_1^T \boldsymbol{X} \, (\boldsymbol{\alpha}_1^T \boldsymbol{X})^T = \boldsymbol{\alpha}_1^T \boldsymbol{R}_{\boldsymbol{X}} \boldsymbol{\alpha}_1 \tag{2.14}$$

The purpose of POD is to maximize the value of the auto-variance tensor of $\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_n$, in a decreasing order of importance to the overall flow. One way to maximize this value is to use Lagrangian multipliers. These multipliers are a strategy to find extremes of a function subject to equality constraints. We have to apply a constraint $\boldsymbol{\alpha}_1^T \boldsymbol{\alpha}_1 = 1$, as to guarantee a maximum. The Lagrangian problem is displayed as:

$$\begin{aligned} &\max_{\boldsymbol{\alpha}_1} \boldsymbol{R}_{\boldsymbol{Y}_1} = \boldsymbol{\alpha}_1^T \boldsymbol{R}_{\boldsymbol{X}} \boldsymbol{\alpha}_1 \\ &\text{subject to } \boldsymbol{\alpha}_1^T \boldsymbol{\alpha}_1 = 1 \end{aligned} \tag{2.15}$$

Which leads to

$$\mathcal{L}(\boldsymbol{\alpha}_1, \lambda_1) = \boldsymbol{\alpha}_1^T \boldsymbol{R_X} \boldsymbol{\alpha}_1 + \lambda_1 (1 - \boldsymbol{\alpha}_1^T \boldsymbol{\alpha}_1) \tag{2.16}$$

A maximum or minimum of a function occurs when its derivative is null. In the case of a Lagrangian multiplier, this derivative is

$$\frac{\partial \mathcal{L}(\boldsymbol{\alpha}_1, \lambda_1)}{\partial \boldsymbol{\alpha}_1} = 2(\boldsymbol{R_X} - \lambda_1 \boldsymbol{I}) \boldsymbol{\alpha}_1 \tag{2.17}$$

For this derivative to be equal to zero,

$$\boldsymbol{R_X} \boldsymbol{\alpha}_1 = \lambda_1 \boldsymbol{\alpha}_1 \tag{2.18}$$

This of course is the equation where $\boldsymbol{\alpha}_1$ and λ_1 are the eigenvectors and eigenvalues of matrix $\boldsymbol{R_X}$.

A better way to understand eigenvectors and eigenvalues is to interpret matrixes as linear transformations. When applying a linear transformation to a vector, if the vector's direction stays unchanged then the vector is an eigenvector of that linear transformation. The eigenvector changes its length by multiplication with a scalar: the eigenvalue associated to that eigenvector.

We must note that, in this case, $\boldsymbol{\alpha}_1$ and λ_1 represent the most important eigenvalue and eigenvector, because they are associated to $\boldsymbol{Y}_1$, the principal component of the vector field. This can also be explained by deducing $\boldsymbol{R_Y} = \boldsymbol{\alpha}_1^T \boldsymbol{R_X} \boldsymbol{\alpha}_1 = \lambda_1$, which brings the conclusion that λ_1 must be as large as possible.

To get the second main component, we arrive at a similar equation to Equation (2.15):

$$\begin{gathered} \max_{\boldsymbol{\alpha}_1} \boldsymbol{R}_{\mathbf{Y}_2} = \boldsymbol{\alpha}_2^T \boldsymbol{R_X} \boldsymbol{\alpha}_2 \\ \text{subject to } \boldsymbol{\alpha}_2^T \boldsymbol{\alpha}_2 = 1; \\ \text{and } \boldsymbol{\alpha}_2^T \boldsymbol{\alpha}_1 = 0 \end{gathered} \tag{2.19}$$

Note that the added constraint exists to guarantee that the eigenvector $\boldsymbol{\alpha}_2$ is orthogonal to $\boldsymbol{\alpha}_1$. This means that the second main component doesn't carry information from the first.

Solving in a similar fashion as before, with an inclusion of a constant u provided by the extra condition, we get

$$2\boldsymbol{\alpha}_1^T (\boldsymbol{R_X} - \lambda_2 \boldsymbol{I}) \boldsymbol{\alpha}_2 + \mathrm{u} = 0 \tag{2.20}$$

Which, knowing that $\boldsymbol{R_X}$ is a symmetric matrix and that $\boldsymbol{\alpha}_1$ is one of its eigenvectors, translates to

$$2\lambda_2 \boldsymbol{\alpha}_2^T \boldsymbol{\alpha}_1 + \mathrm{u} = 0 \tag{2.21}$$

Applying the second constraint established in Equation (2.19), we get $\mathrm{u} = 0$. Ultimately, we get another equation as Equation (2.18), but now we have λ_2 and $\boldsymbol{\alpha}_2$ as eigenvalue and eigenvector. In this case, since the largest eigenvalue is already λ_1, λ_2 corresponds to the second largest eigenvalue.

This procedure can be equally taken for next principal component and the one after that and so on until the nth principal component.

Many authors that have dealt with POD have different notations for the method above described. One of the most important aspects to set clear is that the eigenvectors deduced this way are many times referred to as eigenfunctions, as in other works on POD by Liné [28] and Gabelle [29]. This is probably due to the fact that in this deduction we are only dealing with a set of vectors $\boldsymbol{X}$, when in flow dynamics we are dealing with velocity components in multiple directions, so it may be less confusing to treat these eigenvectors as eigenfunctions so it doesn't get mixed up with velocity vectors and velocity components.

Applying the POD to turbulent flow, instead of having a set of vectors, we can have arrays of flow data that can be calculated using numerical simulation tools such as ANSYS Fluent, where each cell of a mesh has various associated variables. A common analysis focus solely on the velocity field on a 2D flow, where the array represents the velocities in x and y of each cell. To calculate POD effectively in this case, it is needed to reshape each array in a vector, and then combine both in a larger vector. In a transient state, we then get a new array where each column represents a different instant and contains all the velocities of each instant. This array is analogous to the set of vectors x used to explain the math behind the POD.

It is important to note that, in this case, the eigenvectors obtained are like the columns containing all the velocity values. This information is relevant because this means that splitting each eigenvector in two and then reshaping each to the dimensions of the mesh returns also a map of information for each cell. In fact, when applied to only two variables (x and y components of velocity), it is possible to represent these variables in a vector field, similar to the original velocity field.

2.2 Impinging Jets Mixers and NETmix

Over the last decades, CIJs have appeared as an alternative to conventional mixers, and also as well suited mixers for processes that require a rapid homogenization of streams [30]. As explained in the introduction, the jets in a CIJ impinge in the chamber of the mixer and the resulting mixture flows through the mixing chamber runner until the outlet [31].

The first relevant studies in CIJs were done in the 1980's, by Tucker and Lee and have demonstrated that when two opposed jets collide, a highly dynamic flow with chaotic velocity

fluctuations is achieved if the jets are above a certain critical Reynolds number, at the injection area at the top of the chamber, and the flow rapidly becomes parallel and homogenous as it gets further away from this point [32, 33]. This phenomenon allows for a rapid dissipation of energy which is ideal for mixing, providing a quick mixture without any moving parts. The hydrodynamic instability generated from the impingement of the two jets is generated at low Reynolds numbers, $Re = 100$, making CIJs suitable for the mixing of high viscosity liquids [31].

The validation of a 2D model for the study of flow in CIJs was done extensively by Santos *et al*. in recent work [9, 10], as well as the work of Gonçalves *et al* [3, 34].

Regarding the NETmix mixer, the first extensive study was done by Laranjeira in his Ph.D. thesis in 2005 [11] and further worker on the NETmix were published by Laranjeira *et al* in 2009 and 2011 [35, 36] where he describes the mixer as a network model of chambers and channels developed to describe and simulate laminar and turbulent flows and transport phenomena.

Those studies have indicated that, in comparison to other static mixers, the NETmix offers a series of advantages, including a fairly simple design, a big surface for heat exchange (useful for exothermic reactions) and the possibility for scale-up resorting to the association of several mixers in parallel or in a series [11].

A patent was indeed issued to protect this design in 2005 (published in 2009), submitted on request of Faculdade de Engenharia da Universidade do Porto, from the research developed at LSRE – Laboratory for Separation and Reaction Engineering [37].

Gomes' Ph.D. thesis focused more on the mixing component of the NETmix and developed a 2D prototype of the mixer, changing the spherical chambers to cylinders and the circular cross section of the channels to a rectangle, stating that the flow mechanisms present in the regular NETmix were mainly 2D [38].

3 Turbulence in Opposed Jets Mixers

3.1 CIJ Model Description

3.1.1 Geometry

The geometry used for the computational simulations with the CIJs was similar to the 2D one used in Gonçalves work [3], with a rectangular chamber and two opposed injectors near the top of said chamber, and an outlet in the bottom of the chamber. In Figure 4 the geometry is displayed. In this work, the diameter of the chamber, D, is 10 mm, the inlet diameter, d, is 1.5 mm and the length of the chamber, L, is 5 cm.

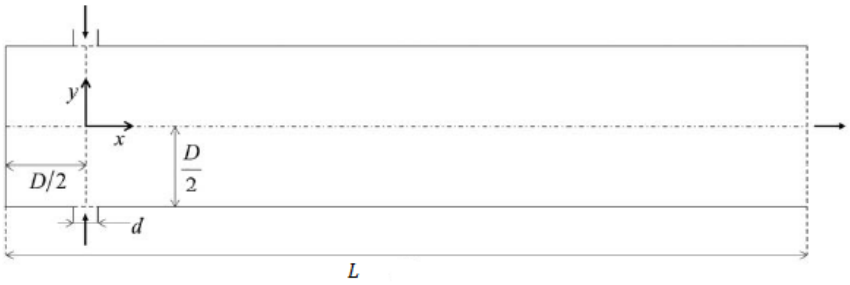


Figure 4 Geometry and dimension of the 2D CIJ, adapted from [3]

3.1.2 Boundary conditions

The simulation was done using the numerical integration of the Navier-Stokes and conservation of mass equations. The Navier-Stokes equation describes the continuity of momentum and has the form

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right) = -\nabla p + \mu\nabla^2\boldsymbol{u} \tag{3.1}$$

Where ρ is the density, $\boldsymbol{u}$ is the velocity, t is the time, p is the pressure and μ is the viscosity.

The conservation of mass equation, for an incompressible flow, is expressed as

$$\nabla\cdot\boldsymbol{u} = 0 \tag{3.2}$$

The chamber and channels walls have a no-slip condition ($\boldsymbol{u} = 0$) throughout the simulation.

The inlet velocity was not uniform at the injectors, so a user defined function (UDF) was implemented in ANSYS Fluent to create a parabolic profile for the velocity, based on

$$u(x) = 1.5\,u_{inj}\left(1 - \left(\frac{x}{d/2}\right)^2\right) \tag{3.3}$$

where u_{inj} is the average velocity at the inlets. The outlet has a gauge pressure of 0, i.e., $p = p_0$.

To study the influence of the Reynolds number, several different injection velocities were used, each corresponding to a different simulation. To calculate each velocity, we use the equation:

$$Re = \frac{\rho u_{inj} d}{\mu} \tag{3.4}$$

in which ρ is the density and μ is the viscosity. Given a Reynolds number, Re, one can solve for the velocity u_{inj}.

In these simulations, the fluid properties used were $\rho = 1000\ \mathrm{kg/m^3}$ and $\mu = 0.02\ \mathrm{Pa \cdot s}$. This viscosity, as noted by Gonçalves [3] is representative of the value used in RIM formulations, albeit in the lower end of the common range of fluids used in the industry. These values of viscosity and density are also present in a large collection of experimental data [10, 31, 39-41].

When simulating transitional flows, the mesh density must be taken into consideration, since smaller the spatial discretization, the more reliable the results. This is related to the number of cells a particle goes through for each time-step calculation, and this number is called the Courant number [42]. Normally, this is the defining factor when choosing the right time-step size, which can be obtained by using the equation

$$\Delta t = \frac{\Delta x}{\frac{3}{2}\bar{u}} \tag{3.5}$$

In which Δt is the time-step, Δx is the size of each cell and $\bar{u}$ is the average velocity of the jets. The $\frac{3}{2}$ factor exists because this is the maximum velocity at the center of the jets. The size of each cell is constant, so in this work the size defined in the meshing process – using ANSYS Meshing - is $62.5\ \mu\mathrm{m}$, as it is the dimension used in Gonçalves work [3]. Considering the velocity as the average velocity in the inlets, u_{inj}, the time-steps for each Reynolds number, along with its velocity, is listed in Table 1.

Table 1 Simulated inlet velocity and time-step size for each Reynolds number in the CIJs

Re	Inlet Velocity $(\mathrm{m/s})$	Time-step size $(\mathrm{s}\times10^{-6})$
250	3.33	6.25
300	4.00	5.21
400	5.33	3.91
500	6.67	3.13

3.2 Probability Density Functions

3.2.1 Methodology

To start the simulation, a steady state was calculated with ANSYS Fluent – which proved to be unachievable with steady but high calculation residuals-, and then 5 residence times of transient state were simulated dynamically, in order to reach a flow solution not influenced by the starting conditions, which could be prejudicial to the first transitional results due to the uncertainty of the steady state calculation..

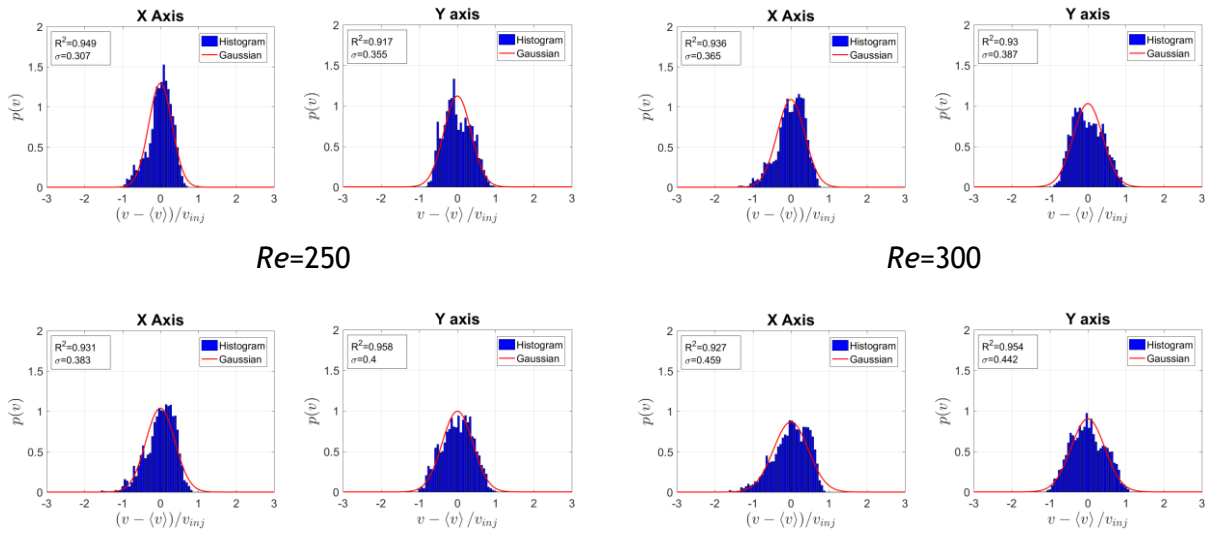
After the initial transient calculation, the velocities were recorded for each time step in a total of 5 residence times for each Reynolds number described in Table 1.

The PDFs calculated are not of the velocities in the y axis and x axis, but of their fluctuating component. Both the fluctuations are divided by the injection velocity, so it is possible to evaluate the change of these variables with the change of injection velocity. It is important to note that a normalization of these variables with their mean would result in abnormal PDFs for he y component velocities, since the mean of these fluctuation is 0 since the two jets collide with the same velocity in opposite directions in the y axis.

The histograms of these distributions are displayed below, in the results, along with a curve representing the normal distribution with the mean and standard deviation of the distribution.

3.2.2 Results

The calculated PDFs of the velocity in the CIJs are displayed in Figure 5, where v represents the velocity in the direction studied.



Re=250 *Re*=300

Re=400 *Re*=500

Figure 5 PDFs for the CIJ mixer

It is clear that both distributions for each Reynolds number are centered at zero, which indicates a good distribution around the mean. For all the distributions of the fluctuations regarding the x axis, their deviation around the mean increases for higher Reynolds numbers. This means that the higher the Reynolds number, the more turbulence intensity is expected from the flow in the CIJs. Regarding the y axis, the distributions also have a higher standard deviation for higher Reynolds numbers, but the increase is not as noticeable.

3.3 Turbulence Integral Scales

3.3.1 Methodology

To calculate the turbulence integral scales, the velocity values of a line throughout the chamber axis up from the jet impact point to the outlet (displayed in Figure 6) were exported for 5 residence times. The values of the y component velocities were subject to a treatment as indicated before in the State of the Art. Only the y component velocities were treated because these fluctuations are the ones that have significance in the formation of vortices.

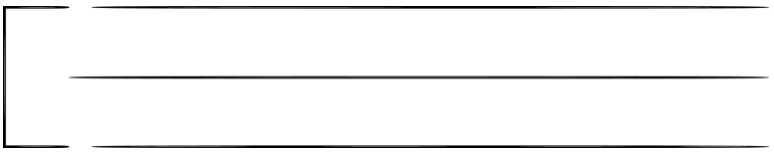


Figure 6 Line in the CIJ from which the velocity data was exported

In detail, a code elaborated in MATLAB treated the data to an autocorrelation. This autocorrelation throughout a line is calculated as:

$$R\,(x) = \frac{u'(x) \times u'^{T}(x)}{N} \tag{3.6}$$

With u' being the velocity fluctuation around the mean for each point of $\boldsymbol{x}$, taken from a total of 4000 time steps, as before. The division by the total number of time steps warrants the average of the correlation of the time steps. The result is a square matrix with the side being the number of coordinate points. The correlation of each point with the rest of the points is then represented with a plot, such as the one shown in Figure 7.

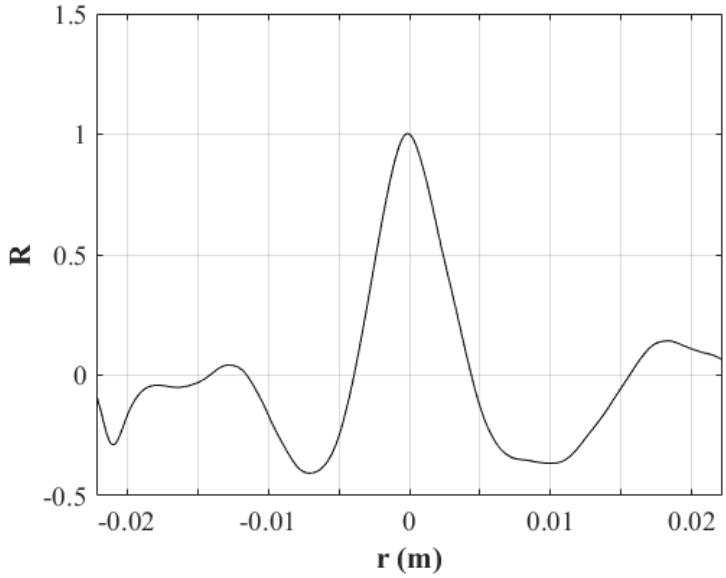


Figure 7 Correlation of the y component velocities in the center point of the line, for Re=300

The turbulence integral lengthscale is the integral between the peak and the zero intercept to the right.

The code used in MATLAB is presented in Appendix C.

The turbulence integral scales were obtained from the data of previous simulations with the Reynolds numbers of 250, 300, 400 and 500.

3.3.2 Results

The turbulence integral lengthscales for the different Reynolds numbers along the chamber axis are shown in Figure 8.

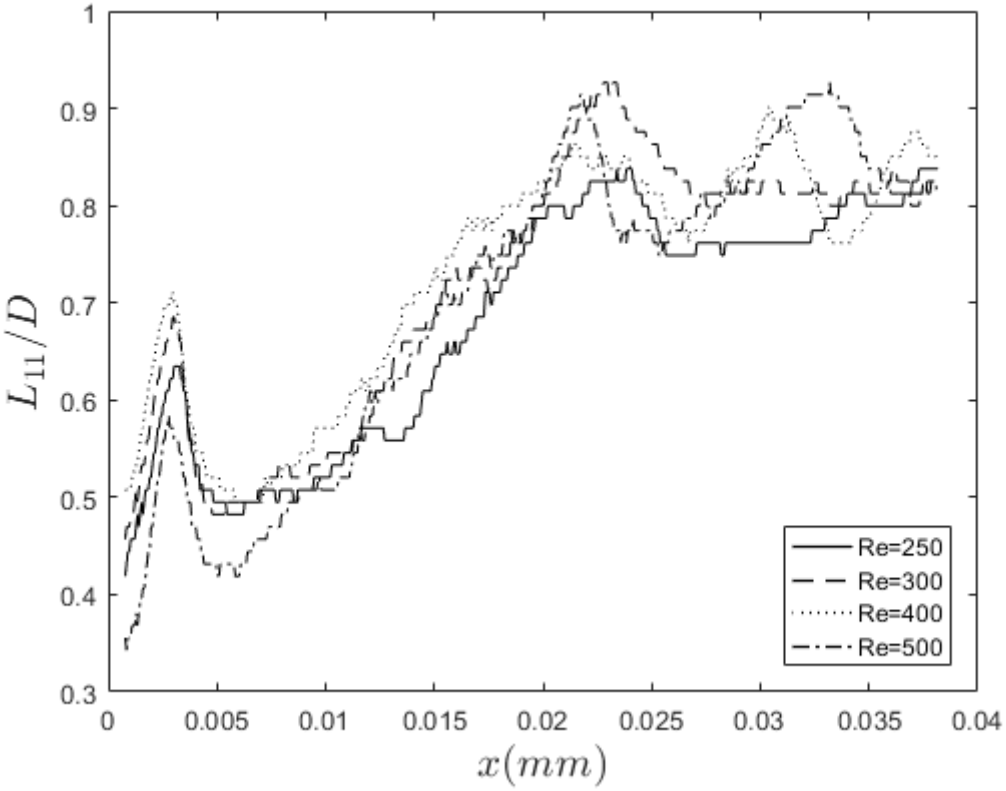


Figure 8 Turbulence Integral Scales of CIJs, for each Re

In comparison with the scales calculated on Gonçalves work [3], the results are similar, verifying an average of around $0.85\times D$, meaning that the eddies formed are restricted by the diameter of the chamber. It can also be noticed that the vortices at around $2.5\ \mathrm{mm}$, which is close to the point where the velocities for the PDF were calculated - the contact point of the jets - are about half the size of the chamber, evolving to larger sizes as they progress towards the outlet.

The full length of the line is not displayed in the graph, because the space left between the last points and the outlet do not allow for an eddy to fully develop, hence making it impossible to have a correlation of 0.

3.4 Turbulence Spectra

A detailed and recent study of the turbulence spectra of the CIJs can be found in Gonçalves work [3], so it won't be repeated on this thesis.

3.5 POD

3.5.1 Methodology

To apply Proper Orthogonal Decomposition to the CIJs, an automatic export function was defined on ANSYS Fluent to export the x and y velocity components in all cell centers of the mesh, throughout 5000 time steps at a Re=300 (after fully developed flow). Since this is a

uniform mesh (all cells have the same size), the velocity field doesn't require any interpolation, only a rearrangement in matrix form. To achieve this, a function translating the ASCII data (exported from the simulation) to various arrays (one for each time step) in MATLAB format was used, and it is shown in Appendix A.

After this, POD was applied using the code explained in Appendix B. In this code, there are some important parts that should be mentioned:

- Since the mesh has a lot of cells (128160), and the total of time steps is 5000, this means that the total of cell elements is $128160 \times 5000 \times 2$ (one for each velocity component, x and y), which results in a major RAM usage, which leads to the need of sampling the matrix. This can be achieved either by dimension skipping (reading only one of every three values, for example) or time step skipping. These are both values that are changeable in the code.
- The correlation matrix, from which the eigenvalues and eigenvectors are taken, is created by creating a matrix that contains all the velocities at all the time-steps (considering the sampling), like so

$$\begin{bmatrix} u_x(x_1,y_1,t_1) & \dots & u_x(x_1,y_1,t_n) \\ \vdots & \ddots & \vdots \\ u_x(x_1,y_n,t_1) & \dots & u_x(x_1,y_n,t_n) \\ \vdots & \ddots & \vdots \\ u_x(x_n,y_n,t_1) & \dots & u_x(x_n,y_n,t_n) \\ u_y(x_1,y_1,t_1) & \dots & u_y(x_1,y_1,t_n) \\ \vdots & \ddots & \vdots \\ u_y(x_1,y_n,t_1) & \dots & u_y(x_1,y_n,t_n) \\ \vdots & \ddots & \vdots \\ u_y(x_n,y_n,t_1) & \dots & u_y(x_n,y_n,t_n) \end{bmatrix}$$

  Multiplying then the matrix by its transpose. This is the step that requires the most RAM, as it is an operation that uses three large arrays.

There are two options to calculate the eigenvalues and eigenvectors. The *eig()* function is the standard one, that retrieves all the eigen properties and works on any matrix. Another option is the *eigs()* function, that only returns the largest eigenvalues (and their associated eigenvectors) and only operates on *sparse* matrices. A sparse matrix is a matrix in which most of the elements are zero, that can be stored in MATLAB as a series of values and their coordinates, instead of an array with all of its components. It is particularly useful in matrices that have a lot of zeroes, since these values are not stored in a sparse matrix. Also, the number of eigenvalues returned by this function is defined by the user, the default being 6; in this case, we used 100. Both functions will be tested to analyze their differences.

Initially the following conditions were set:

- Dimension skip of 3 and no time skip, using *eig()*.
- Dimension skip of 3 and time skip of 20, using *eig()*.
- Dimension skip of 3 and no time skip, using *eigs()*.
- Dimension skip of 3 and time skip of 20, using *eigs()*.

Besides plotting some results that are in the section below, some data was saved for each experiment. The data saved contains the eigenvectors matrix, the eigenvalues matrix, the number of time steps, the dimensions of the mesh, the sum of the eigenvalues and the (sampled) velocities matrices and dimension vectors.

3.5.2 Results

The results of each application of the code, in terms of computational time and data storage size, is shown in Table 2.

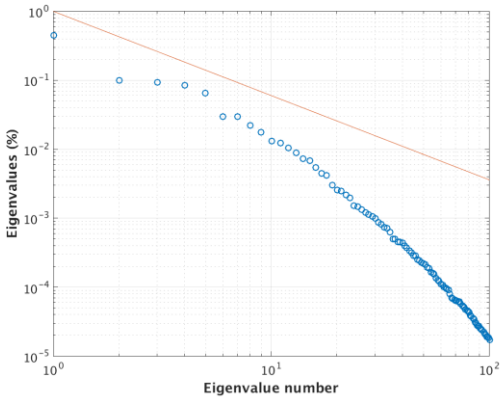
Table 2 Computational time and data storage size of each run

Condition	Computational time (s)	Data Storage Size (MB)
eig(), Dimension skip = 3 Time Skip = 1	1606	7388
eig(), Dimension skip = 3 Time Skip = 20	2241	6349
eigs(), Dimension skip = 3 Time Skip = 1	320	1117
eigs(), Dimension skip = 3 Time Skip = 20	276	76

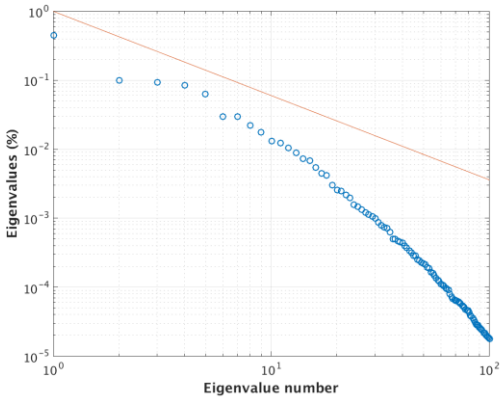
It is possible to see that the use of *eigs()* not only reduces significantly the computational time, as it also reduces the space occupied by the data. This happens because the number of eigenvalues calculated and stored is significantly lower.

One important result to take out of this procedure is the magnitude of the eigenvalues, because this indicates which eigenvectors are the most important to define the flow. To do this, one uses a POD eigenvalue spectrum, which plots the eigenvalue versus the number of the mode associated with them. According to Knight and Sirovich [43], these spectra show a trend with a slope of $-11/9$, characteristic of the inertial subrange of turbulence. To compare the effect

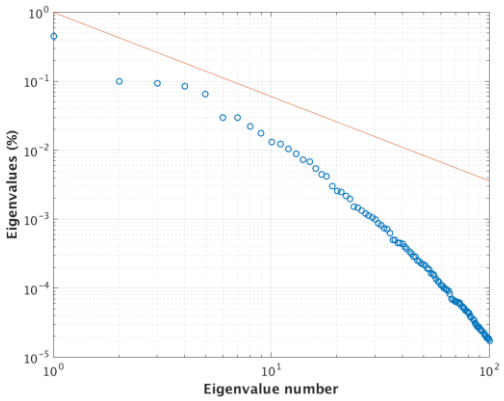
of the different function and dimension skip, the spectra of the eigenvalues on the different conditions are shown in Figure 9.



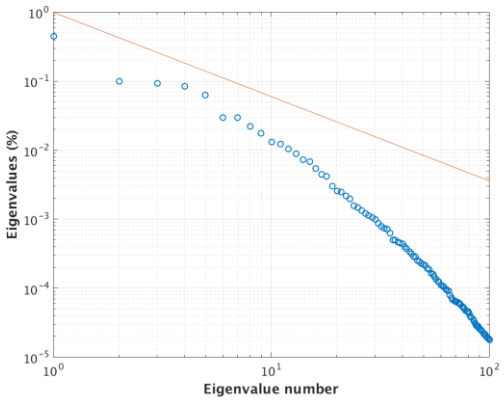
eig(); Dimension skip = 3; Time Skip = 1



eig(); Dimension skip = 3; Time Skip = 20



eigs(); Dimension skip = 3; Time Skip = 1



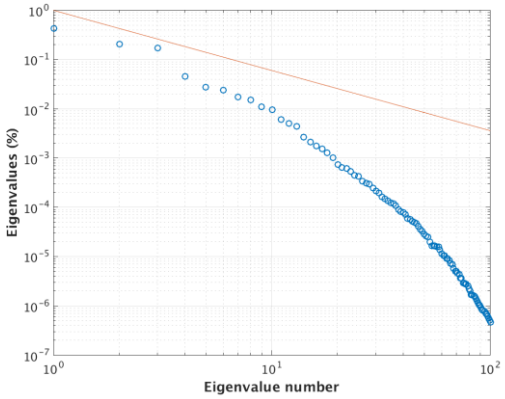
eigs(); Dimension skip = 3; Time Skip = 20

Figure 9 Comparison of the eigenvalue spectra for each condition

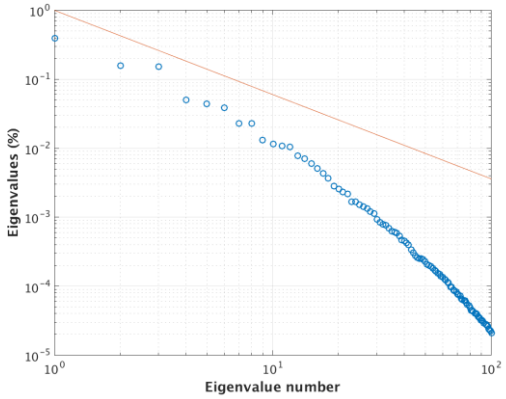
It is possible to see that there is little to no impact on the spectra with the different conditions.

This enabled to procure a better analysis of the flow, by extending the number of time steps to 40000 (which is equal to 5 residence times of the reactor). To reduce data storage requirements, and concluding that a time skip of 20 is not impactful, only 1 out of every 10 time steps was exported and treated with the same procedure. Instead of changing the function, this time the variable in focus was the Reynolds number of the flow.

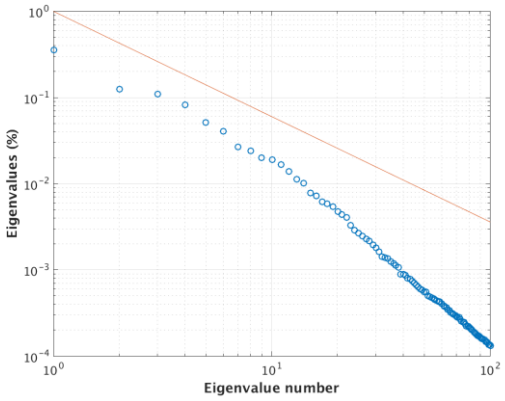
Before proceeding to the results of the POD on this data, the sampling of the range was also tested, trying different numbers of timesteps. The eigenvalue spectra on these conditions are shown in Figure 10.



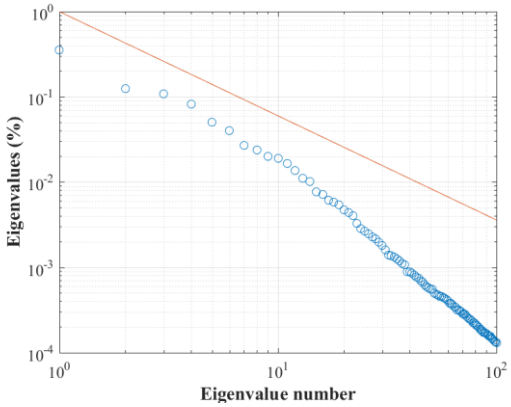
400 time steps



1000 time steps



2000 time steps

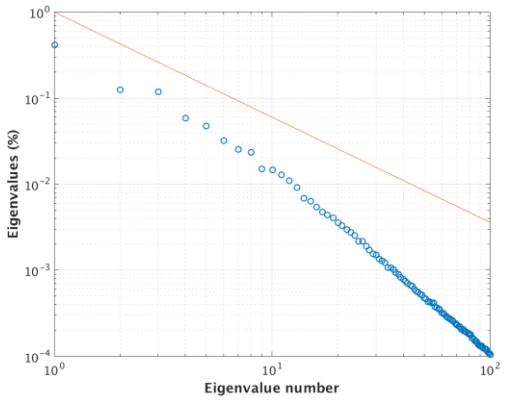


4000 time steps

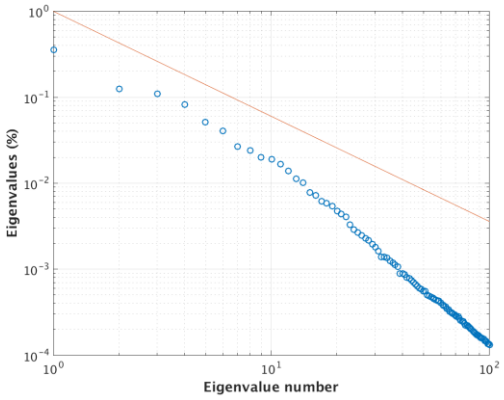
Figure 10 Eigenvalue spectra of different ranges

It is possible to see that the larger the number of time steps, the more similar are the results to one another. So, it didn't prove useful to shorten the range, as the results wouldn't be as correct as the results with the full range.

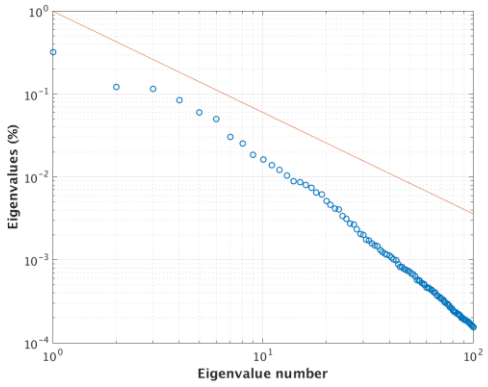
The eigenvalue spectra of the POD with the full range of time steps exported are shown in Figure 11, where the eigenvalues are normalized, by dividing each by the sum of all.



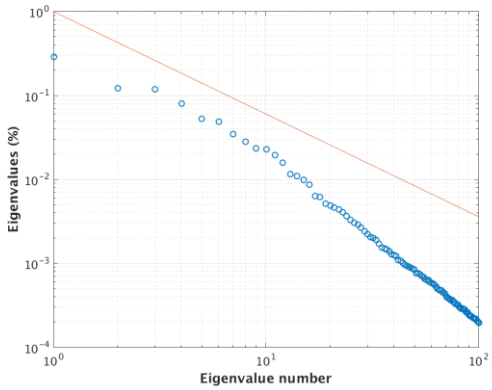
Re=250



Re=300



Re=400



Re=500

Figure 11 POD eigenvalue spectra for the different Reynolds numbers simulated in CIJs

The sum of all eigenvalues is the total kinetic energy [26], so this means that the 1st mode is clearly the one that contributes the most to the total kinetic energy, with values close to 40% for all Reynolds numbers. It is also apparent that the 2nd and 3rd modes are very close to each other in terms of magnitude, with values around the percentile. For the following modes, the contribution gets subsequently lower, although modes 4 and 5 have often contribution between 0.5% and 1%. According to Liné, these modes are related to the contributions of different aspects of the flow. The 1st mode, being the largest in magnitude, is the mean flow contribution, modes 2, 3 and sometimes 4 and 5 are characteristic of the oscillating nature of the flow, defined by structures named by Liné as *organized motion*, and the upper modes are what defines turbulence [26]. To better understand what these modes effectively are, the first 4 modes of the POD of the CIJs with *Re*=300 is shown in Figure 12.

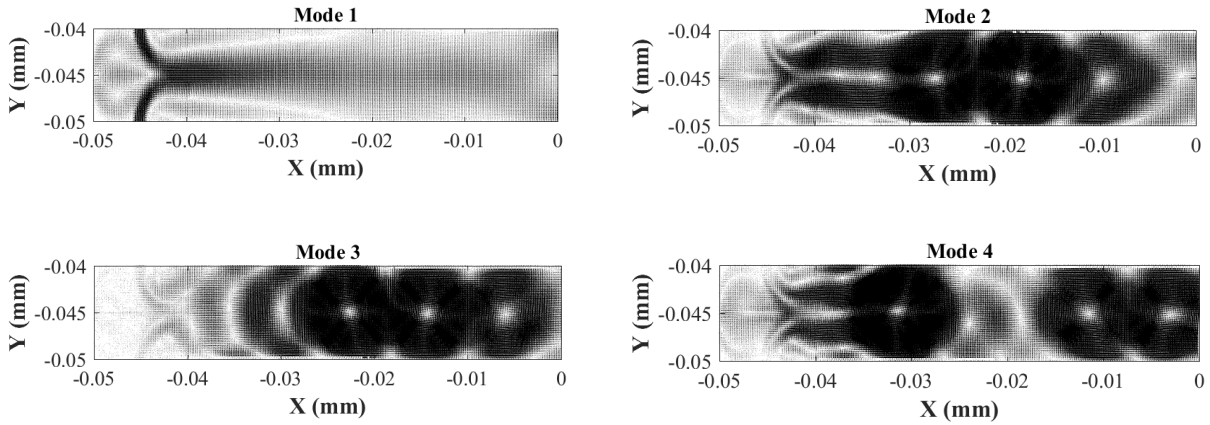


Figure 12 The first 4 modes of the POD of a CIJ with a Reynolds number of 300

While the 1st mode clearly depicts what one would expect to be the mean flow, the following modes show more what seems like the variations of velocity in the many vortices that appear throughout the chamber.

An important application of the POD is the reconstruction of the velocity field regarding the magnitude of each mode calculated. The several modes calculated may seem velocity fields, but eigenvalues and eigenvectors are dimensionless. In order to reconstruct the instantaneous velocity field, one has to not only multiply this eigenvector by a variable regarding dimension, but also regarding time. This variable is known as the POD coefficient, represented by a_k^i, where i is the number of the mode and k is the number of the instant (i.e. the snapshot at a given time). These coefficients are calculated from the original velocity fields [26], as such:

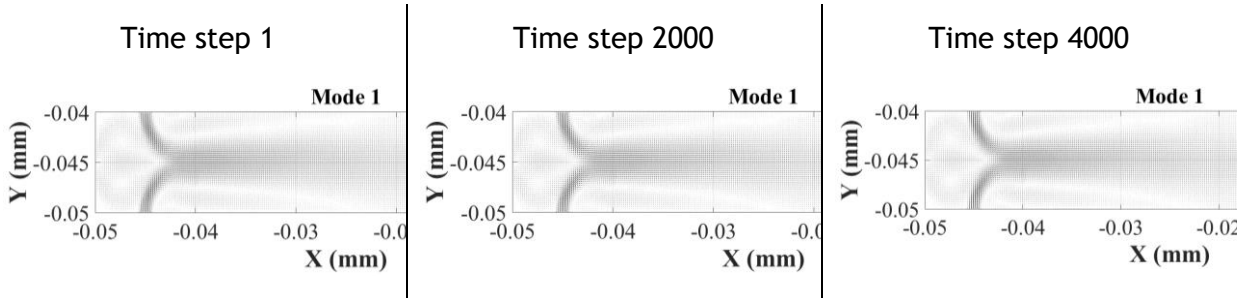
$$a_k^i = \boldsymbol{U}_k \cdot \boldsymbol{\alpha}_\mathrm{i} \tag{3.7}$$

The reconstruction is then made by

$$\widetilde{\boldsymbol{U}}_k = \sum_{i=1}^{\mathcal{N}} a_k^i \boldsymbol{\alpha}_i \tag{3.8}$$

with $\mathcal{N}$ as the number of modes. The more modes used, the most complete the reconstruction is.

This was done for various instants and a video was compiled. The results are synthetized in



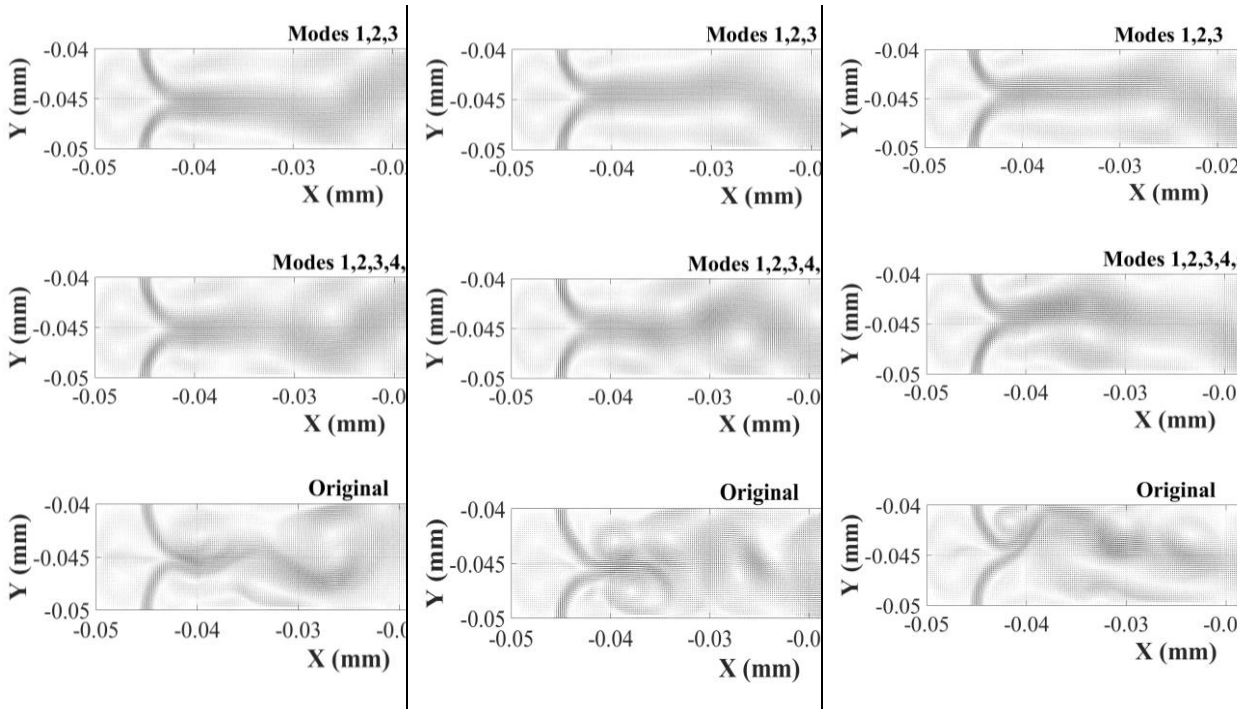


Figure 13.

As expected, the higher number of modes provides a better reconstruction of the original field. As the number increases, the existence of vortices becomes also increasingly more apparent, which is the main focus of the POD: with only 5 modes one can detect the largest flow structures formed in the flow.

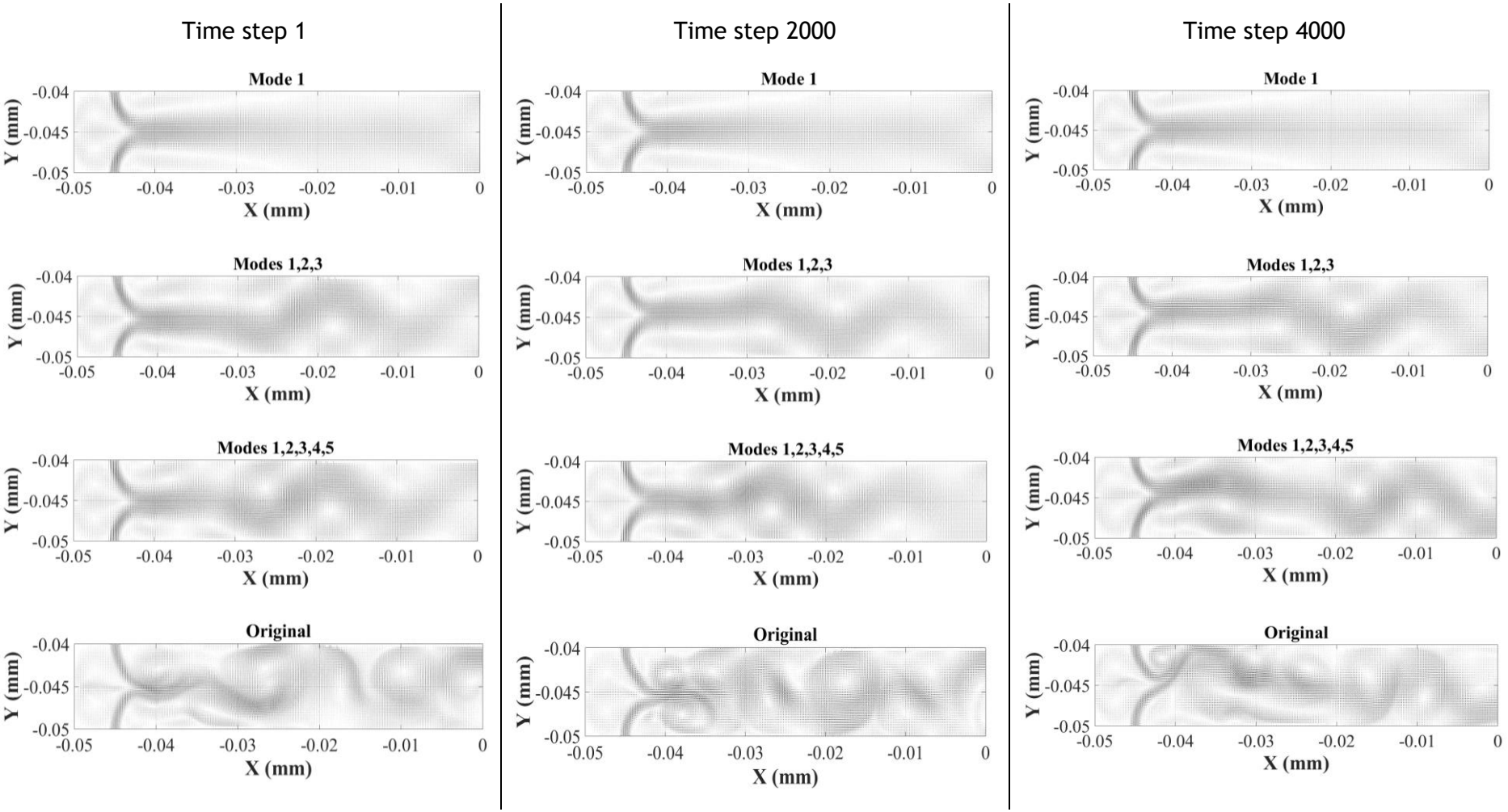


Figure 13 Instantaneous reconstructions and the original velocity field of the CIJ at Re=300. The first column is for the first time step (the start of the data time period), the second column is the middle of the time period and the last column is at the end of the time period

4 Turbulence in NETmix

4.1 NETmix Unit Block Model Description

One of the main issues of the numerical simulation of a NETmix reactor is the dimension and complexity of the reactor. As each reactor has many cells, each with complex coherent structures inside, the computing time to simulate the hydrodynamics of a full reactor is an obstacle in the field of mixing study. To overcome this obstacle, the concept of periodic boundary conditions is applied to the NETmix in several different geometries.

The software used, ANSYS Fluent, treats the flow at a periodic boundary as though the opposing periodic plane is a direct neighbor to the cells adjacent to the first periodic boundary [44].

4.1.1 Geometry

The NETmix Unit Block (NUB) consists of a single NETmix chamber with channels to another four halves, where periodic boundary conditions where applied. So, in theory, it represents three lines and an infinite number of columns of a NETmix reactor. See Figure 14 for an illustration of the NUB.

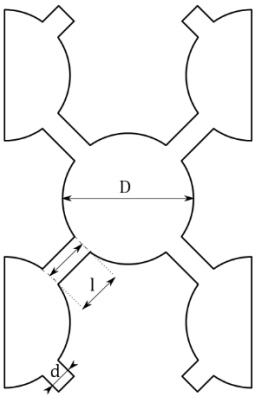


Figure 14 NETmix Unit Block (only one chamber)

In the geometry used for meshing, the chamber diameter, *D*, is 6.5 mm, the channel diameter, or the width, *d*, is 1 mm and their length, *l*, is 2 mm. For the inlets and outlets of the NUB, their dimension is half of the total length of the channels. The mesh cell dimension was also established at $62.5\ \mu m$, as for the CIJs.

4.1.2 Boundary Conditions

Similarly to the procedure used before in the CIJs, various Reynolds numbers were used in these simulations, with a slightly different Reynolds number formula

$$Re = \frac{\rho u_{inj} d_h}{\mu} \tag{4.1}$$

In which ρ is the density, d_h is the hydraulic diameter of the channel and μ is the viscosity. Given a Reynolds number, Re, one can solve for the velocity u.

In these simulations, the fluid properties used were the default for liquid water at 20 ºC in the Fluent database, with $\rho = 998.2\ \mathrm{kg/m^3}$ and $\mu = 0.001003\ \mathrm{Pa \cdot s}$.

The hydraulic diameter is calculated by

$$d_h = \frac{2 \cdot w \cdot d}{w + d} \tag{4.2}$$

where d is the width (the diameter in this case) and w is the depth of the channel. Since this is a two-dimensional geometry, the depth of the channel is immeasurable, so it is considered that the hydraulic diameter is the limit of this expression as w approaches infinity, i.e., $2d$. This means that, in a NETmix channel, $d_h = 2d$. Knowing this, and the properties of the fluid, one can calculate the inlet velocities.

Considering the medium velocity as the velocity at the inlets, the time-steps for each Reynolds number, along with its velocity, are listed in Table 3.

Table 3 Simulated inlet velocity and time-step size for each Reynolds number in the NUB

Re	Inlet Velocity $(\mathrm{m/s})$	Time-step size (ms)
1000	0.502	0.0415
2000	1.01	0.0207
4000	2.01	0.0104
10000	5.02	0.00415

To obtain a segregated flow for the transient simulations, steady-state simulations were done for each Reynolds number, where no-shear walls were created in the middle of each chamber and at the periodic boundaries, this makes it so that the x and y components of the shear stress at the walls are equal to zero, so $\tau_x = \tau_y = 0$.

This strategy is based on a similar strategy used by Fonte in his Ph.D. thesis [45].

4.2 Probability Density Functions

4.2.1 New Geometries and Methodology

To study if the geometric model was adequate for the study of turbulent flow, it was needed to develop different geometries to see how much the flow dynamics would change. It is mandatory, in a study of this kind, to assure that the turbulent flow is fully developed, which is indicated by a PDF that is similar to a Gaussian curve, and in a single chamber, like the geometry shown in Figure 14, this may not happen.

The geometry shown in Figure 14 is now known as *SmallNUB*. The following geometries were created: *NormalNUB* (2 chambers), *DoubleNUB* (4 chambers), *SideNUB* (4 chambers with an extra column at each side) and *BigNUB* (8 chambers). The mesh of each one is shown in Figure 15, Figure 16, Figure 17, Figure 18 and Figure 19. These figures are flipped on their side for better page fitting.

To better organize the data from different chambers, each was assigned a number, correspondent to their row, in ascending order. In Figure 19, the numbers are displayed.

After these geometries were created, a point was created in ANSYS Fluent on the center of each chamber (in the central column), where the velocities in the x axis and the y axis were recorded throughout each time step in a total of 5 residence times for each geometry, for each Reynolds number described in Table 3, in a transitional state. Before this, 5 residence times were taken to assure a fully developed flow.

After taking these velocities, the software MATLAB was used to calculate and plot the PDFs for each chamber of each geometry, at different Reynolds numbers.

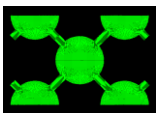


Figure 15 SmallNUB

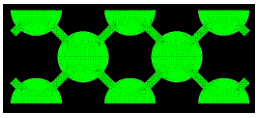


Figure 16 NormalNUB

Figure 17 DoubleNUB

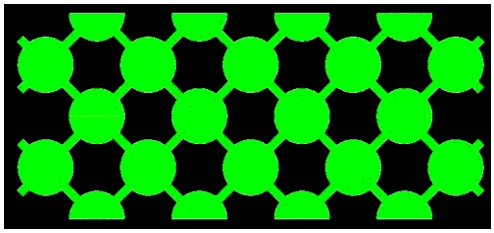


Figure 18 SideNUB

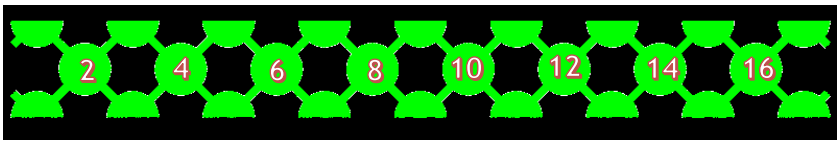


Figure 19 BigNUB with chamber numbering

It is important to note that the PDFs calculated are *not* of the velocities in the x and y axis, but instead in a new axis defined by the outlets orientation. Since the outlets are at 45º and -45º degrees from the central point of the chamber, these velocities are defined as

$$u_{right} = \frac{1}{\sqrt{2}}(u_y + u_x) \tag{4.3}$$

For the right outlet oriented velocity, where u_x and u_y are the velocity components in the x axis and in the y axis, respectively. The left outlet oriented velocity is defined by

$$u_{left} = \frac{1}{\sqrt{2}}(u_y - u_x) \tag{4.4}$$

The code creates PDFs based on the normalized fluctuation of velocity around the injection velocity, as it is here

$$v' = \frac{u - \langle u \rangle}{u_{inj}} \tag{4.5}$$

This enables us to calculate the turbulence intensity of the flow from the standard deviation of the PDFs. The turbulence intensity is calculated by

$$\mathcal{I} = \frac{\sqrt{\overline{(u - u_{inj})^2}}}{\langle u \rangle} = \frac{\left[\frac{1}{T}\int_0^T (u - u_{inj})^2 \, dt\right]^{\frac{1}{2}}}{\langle u \rangle} \tag{4.6}$$

where T is the data series period. If we consider $\overline{v'} = 0$, the formula for the standard deviation is

$$\sigma = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(v' - \overline{v'})^2} = \sqrt{\frac{1}{M u_{inj}^2}\sum_{i=1}^{M}(u - \langle u \rangle)^2} = \frac{\sqrt{\frac{1}{M}\sum_{i=1}^{M}(u - \langle u \rangle)^2}}{u_{inj}} \tag{4.7}$$

Adjusting Equation (4.7) we get

$$\frac{\sigma \cdot u_{inj}^2}{\langle u \rangle^2} = \frac{\sqrt{\frac{1}{M}\sum_{i=1}^{M}(u - u_{inj})^2}}{\langle u \rangle} \tag{4.8}$$

Considering that M is the total number of time steps, then σ is proportional to $\mathcal{I}$.

4.2.2 Results

The PDFs of all geometries are displayed in Figure 20, Figure 21, Figure 22, Figure 23, Figure 24 and Figure 25, where v represents the velocity in the direction studied.

The PDFs are plotted with a Gaussian distribution curve in red, which enables to check if the distributions are closer to a turbulent profile. It is only for the DoubleNUB's geometry that a more uniform distribution of velocity fluctuations is achieved, which indicates a better, most reliable approach to the operating principle of the NETmix.

It is concluded that this geometry is a good approach to the problem, since both SideNUB's and BigNUB's PDFs don't differ significantly.

The results of the evolution of the standard deviation and, therefore, turbulence intensity with the chamber number for the DoubleNUB geometry are shown in Figure 26. It is clear that the higher the Reynolds number, the higher the standard deviation is achieved, and it is also clear that the maximum standard deviation is achieved in less space travelled by the flow. At *Re*=10000 the standard deviation is already constant, whereas at *Re*=1000 the standard deviation at chamber 8 is double the existent at chamber 2.

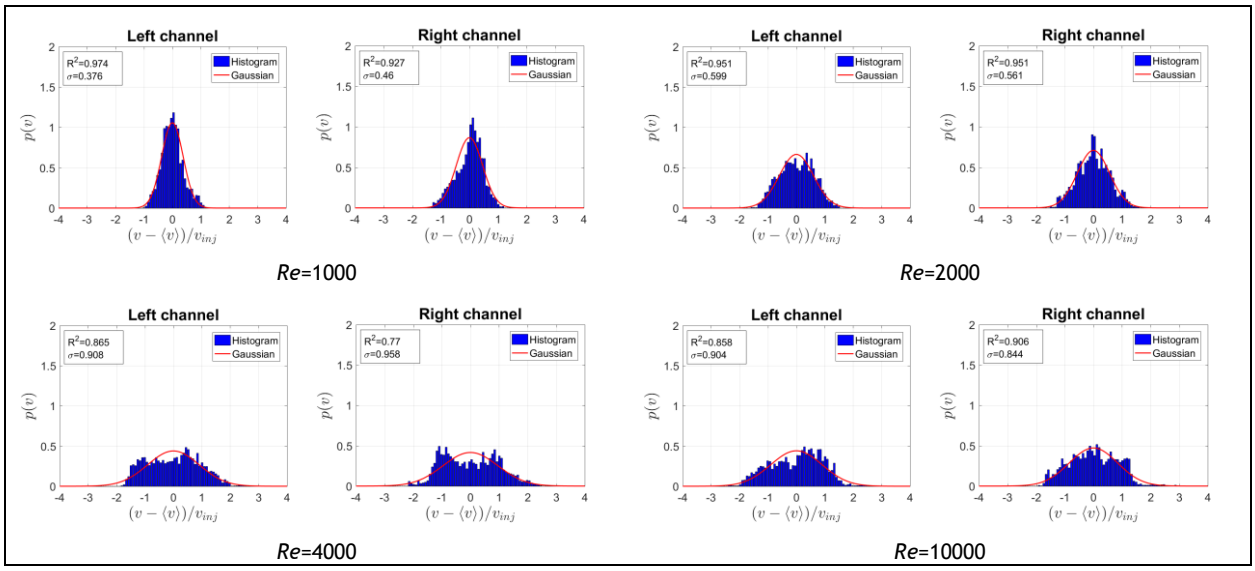


Figure 20 SmallNUB's PDFs

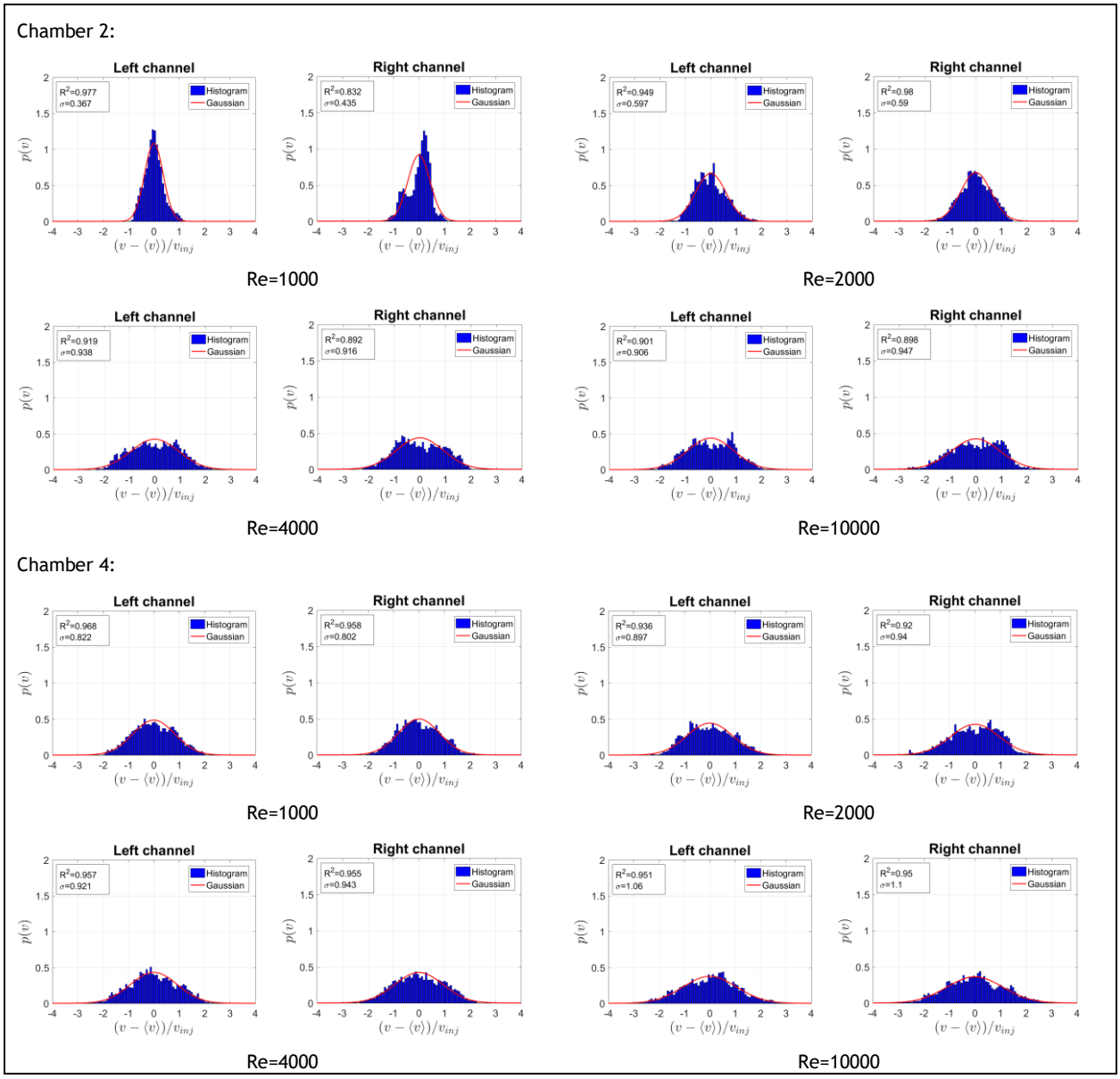
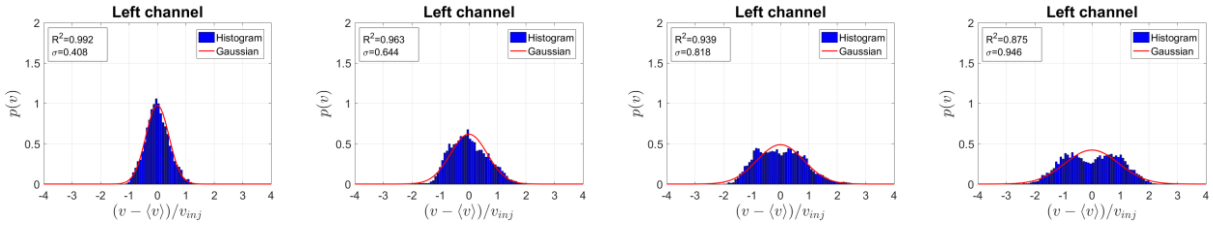


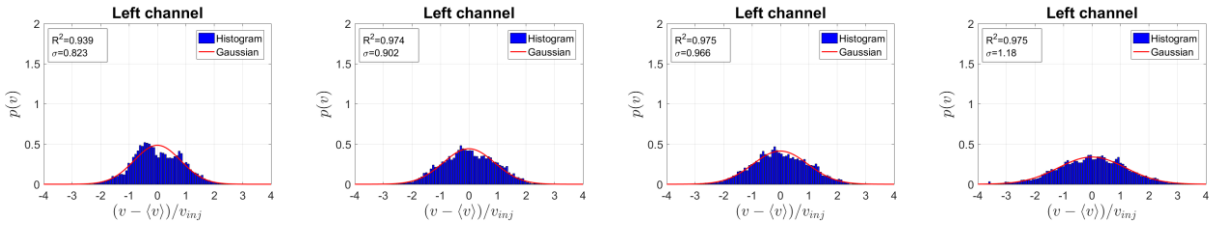
Figure 21 NormalNUB's PDFs

Chamber 2:



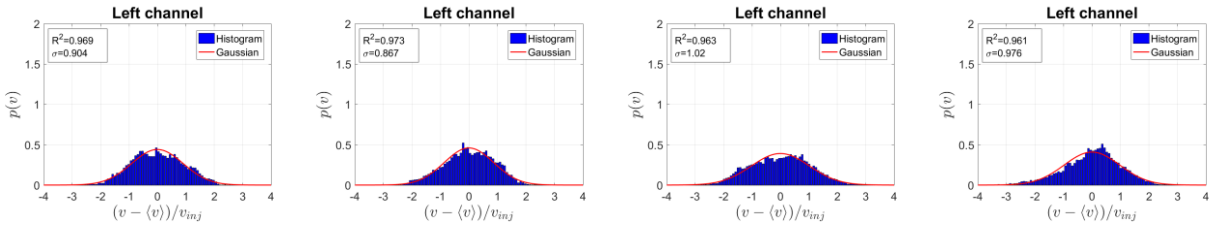
Re=1000 Re=2000 Re=4000 Re=10000

Chamber 4:



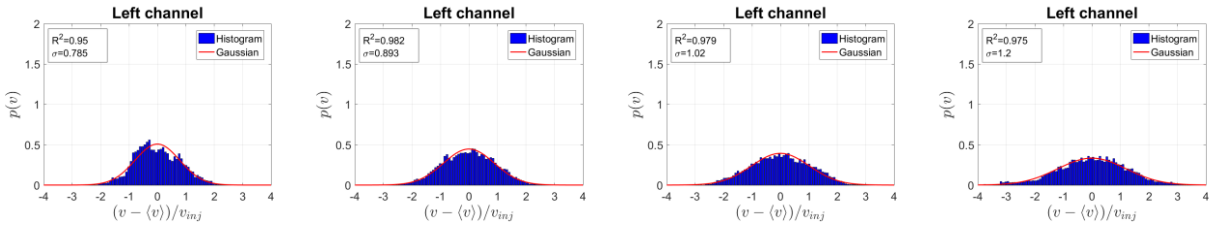
Re=1000 Re=2000 Re=4000 Re=10000

Chamber 6:



Re=1000 Re=2000 Re=4000 Re=10000

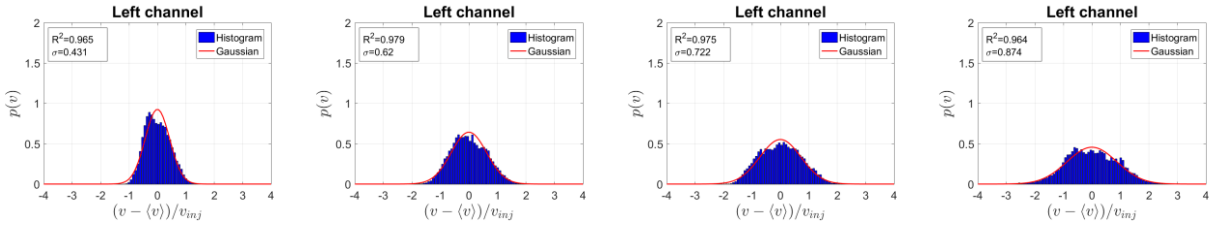
Chamber 8:



Re=1000 Re=2000 Re=4000 Re=10000

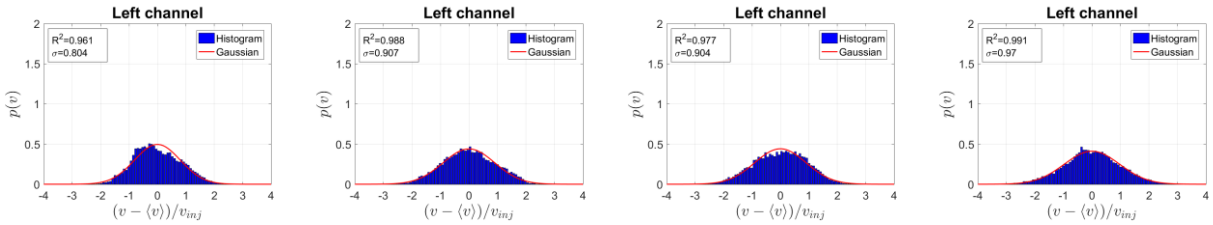
Figure 22 DoubleNUB's PDFs, for only the velocity oriented toward the left channel

Chamber 2:



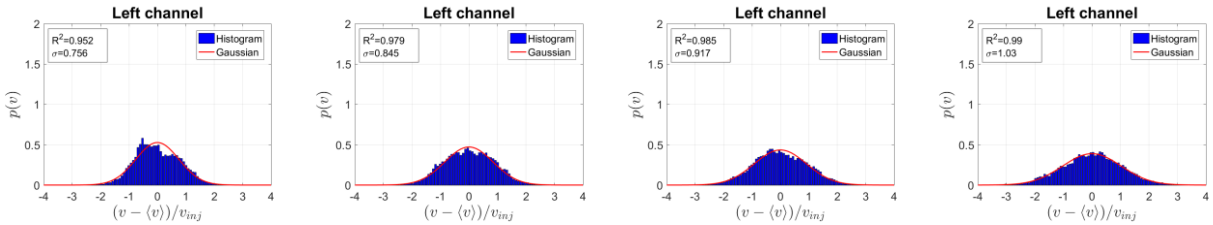
Re=1000 Re=2000 Re=4000 Re=10000

Chamber 4:



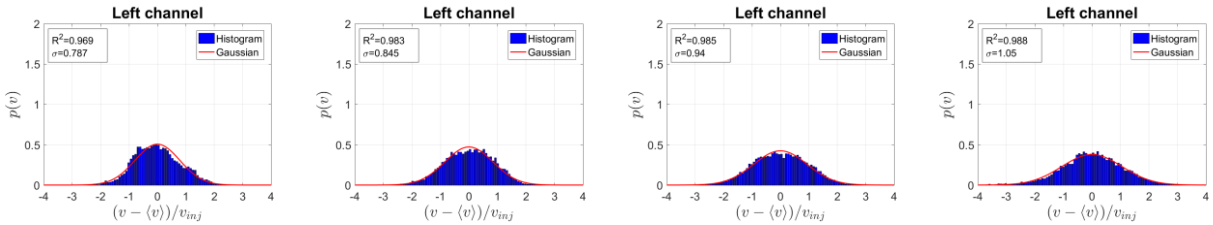
Re=1000 Re=2000 Re=4000 Re=10000

Chamber 6:



Re=1000 Re=2000 Re=4000 Re=10000

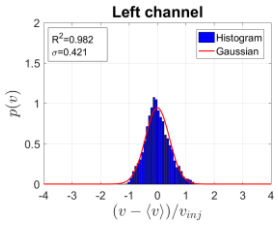
Chamber 8:



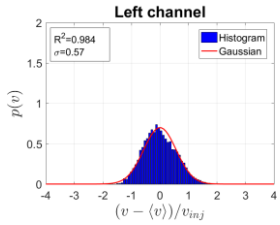
Re=1000 Re=2000 Re=4000 Re=10000

Figure 23 SideNUB's PDFs, only the velocity oriented toward the left channel

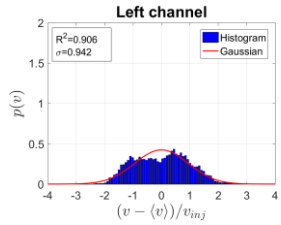
Chamber 2:



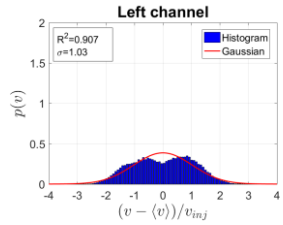
Re=1000



Re=2000

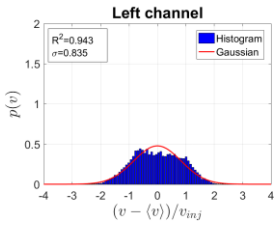


Re=4000

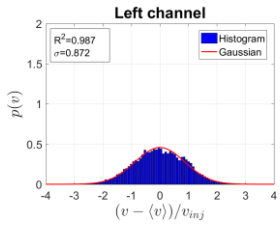


Re=10000

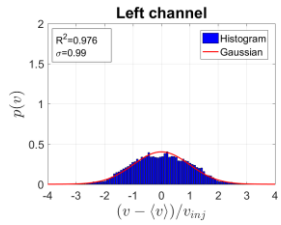
Chamber 4:



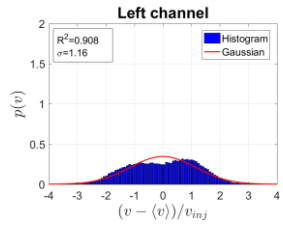
Re=1000



Re=2000

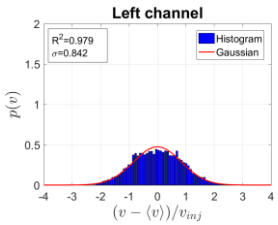


Re=4000

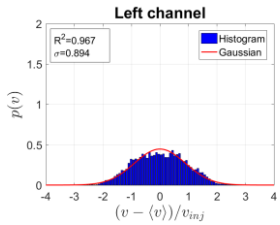


Re=10000

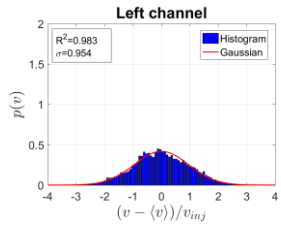
Chamber 6:



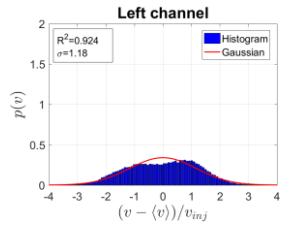
Re=1000



Re=2000

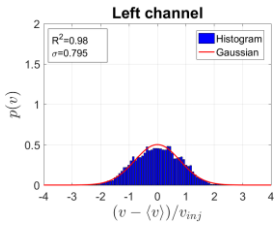


Re=4000

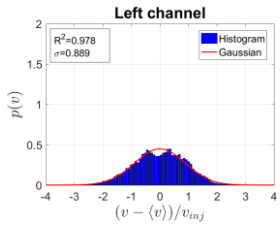


Re=10000

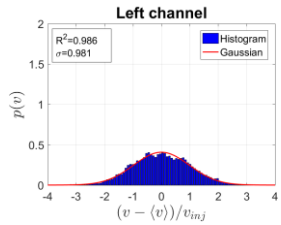
Chamber 8:



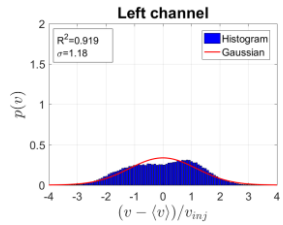
Re=1000



Re=2000

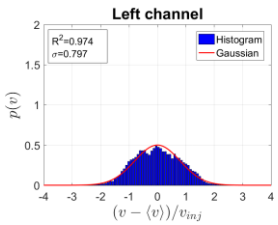


Re=4000

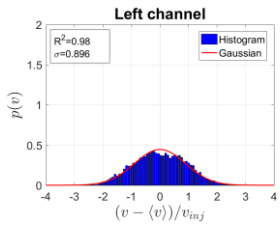


Re=10000

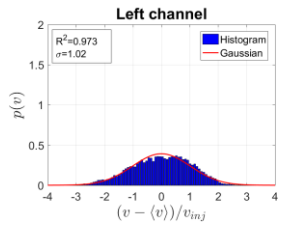
Chamber 10:



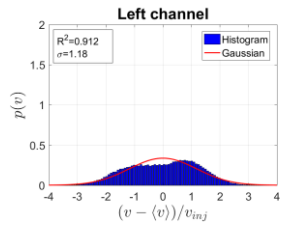
Re=1000



Re=2000



Re=4000



Re=10000

Figure 24 BigNUB's PDFs, only the velocity oriented toward the left channel, chambers 2 through 5

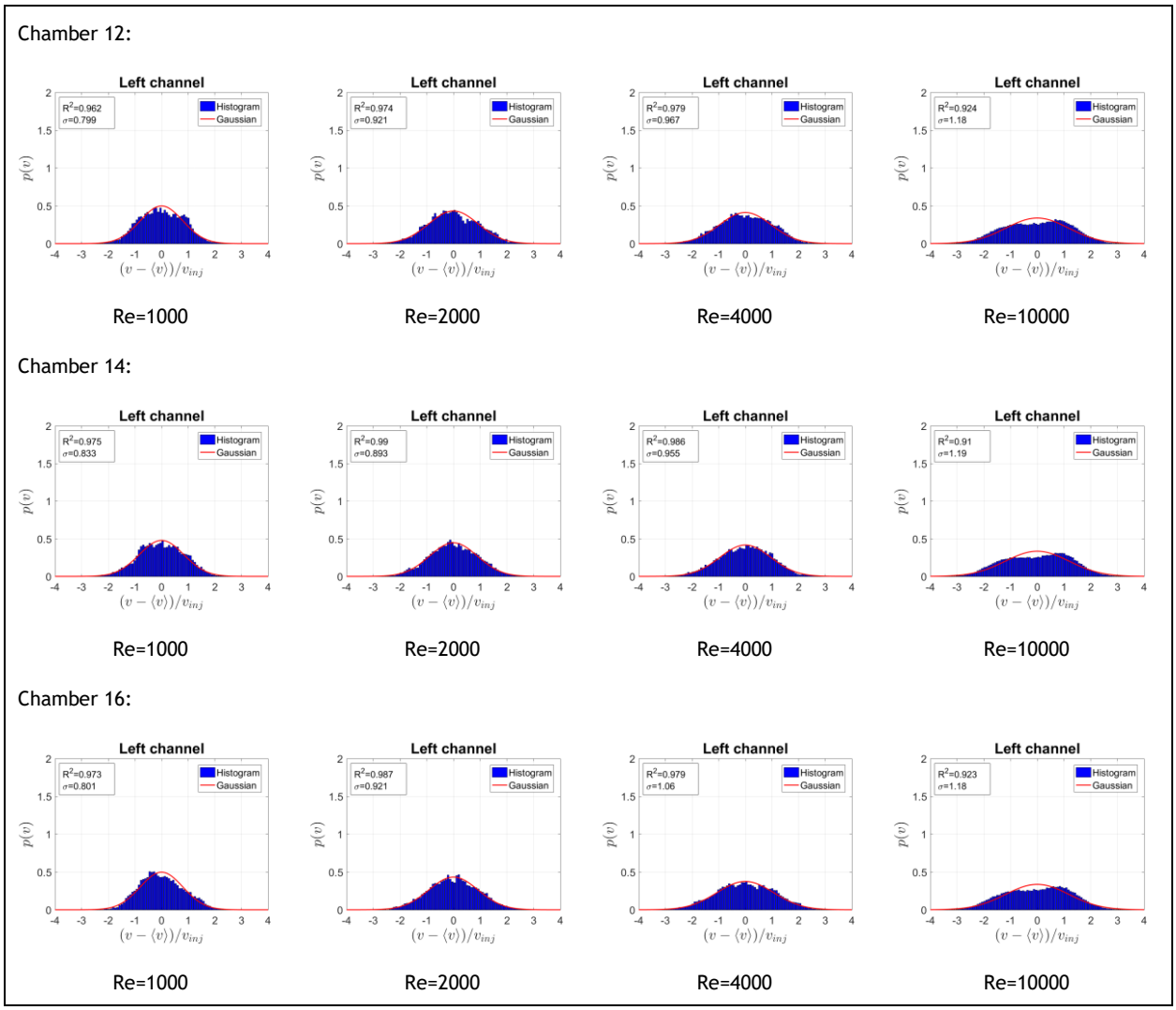


Figure 25 BigNUB's PDFs, only the velocity oriented toward the left channel, chambers 12 through 16

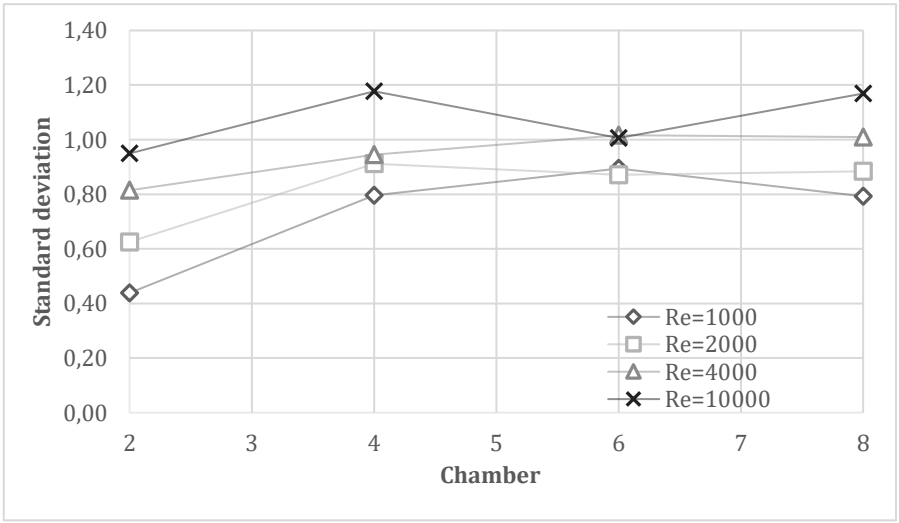


Figure 26 Standard deviation vs chamber number, in the DoubleNUB

4.3 Turbulence Integral Scales

4.3.1 Methodology

To calculate the lengthscales in the NUB, it was necessary to define a line in a chamber to evaluate the autocorrelation of the velocities. A good chamber to do this is chamber 6 in the DoubleNUB geometry. Since this geometry, the flow achieves a velocity PDF that describes a turbulent flow, as explained before, and this chamber in particular guarantees a big enough distance from the inlets, so to give space for the flow to develop fully, and it is also not affected by the boundary conditions existent in chamber 8, where the outlets are located.

The line from which the data was exported from the simulation was in fact a series of 100 points, with equal distance from each other, allocated diagonally in the chamber, as depicted in Figure 27. The line only includes the space inside the chamber and not the channel, as the aim of our study are the scales inside the chamber.

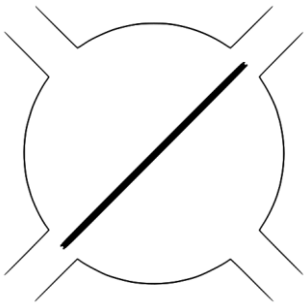


Figure 27 Line in NUB's chamber 6 *from which the velocity data was exported*

From these velocities, the scales were calculated regarding the perpendicular velocity (towards the left outlet). The velocity values were exported for a period of 5 residence times. The procedure was the same as in subsection 3.3.1.

4.3.2 Results

The results are shown in graph form in Figure 28.

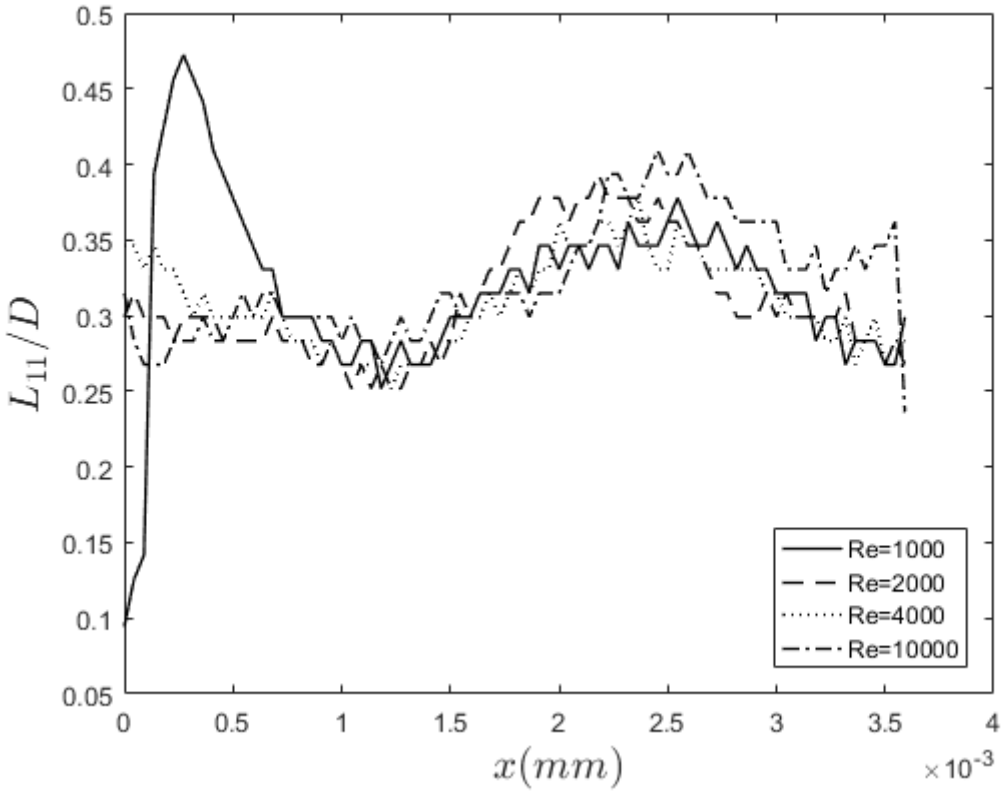


Figure 28 Turbulence integral scales for different Reynolds numbers for the DoubleNUB on its sixth chamber

For lower Reynolds numbers, there is a formation of eddies of almost half the diameter of the chamber, possibly since there isn't a lot of intensity in the oscillation that creates smaller vortices (as seen for higher Reynolds numbers), essentially allowing time for the eddies to form and space between the contact point of the jets and the walls of the chamber to have single large eddies inside.

Similar to what happened in the CIJs, it is impossible to get the full length of the line correlated. It is possible to notice also that the larger eddies for all other Reynolds numbers (except 1000) form near the middle of the chamber, at around a third of the chamber diameter.

4.4 Turbulence Spectra

4.4.1 Methodology

The values used in the elaboration of the PDFs were also used to compute the turbulence spectra. The values used are only the ones of the sixth chamber of the DoubleNUB's geometry, chosen because it is far enough from the inlets to show a fully developed turbulent flow (seen in the PDFs), and the last chamber, although even further, is affected by the boundary conditions at the outlets. It was also found pertinent to study the spectra not only regarding velocities in the inlet axis, but also the velocities in the x and y axis.

The power spectra is obtained from the modulus of the Fourier transforms computed using MATLAB. The power spectra was filtered with a Hammond window function, W, to eliminate the noise in the spectra, E [3]:

$$E_{filtered}(\lambda_i) = \frac{\sum_{j=-\frac{n-1}{2}}^{\frac{n-1}{2}} W(j)E(\lambda_{i+j})}{\sum_{j=-\frac{n-1}{2}}^{\frac{n-1}{2}} W(j)}, W(j) = \xi - \beta\cos\left(\frac{j}{n-1}2\pi\right) \quad (4.9)$$

4.4.2 Results

The results, with 5 residence times as the data acquisition period, for a Reynolds number of 300, are displayed in Figure 29. The plots have an arrow indicating the velocity component used in the power spectra.

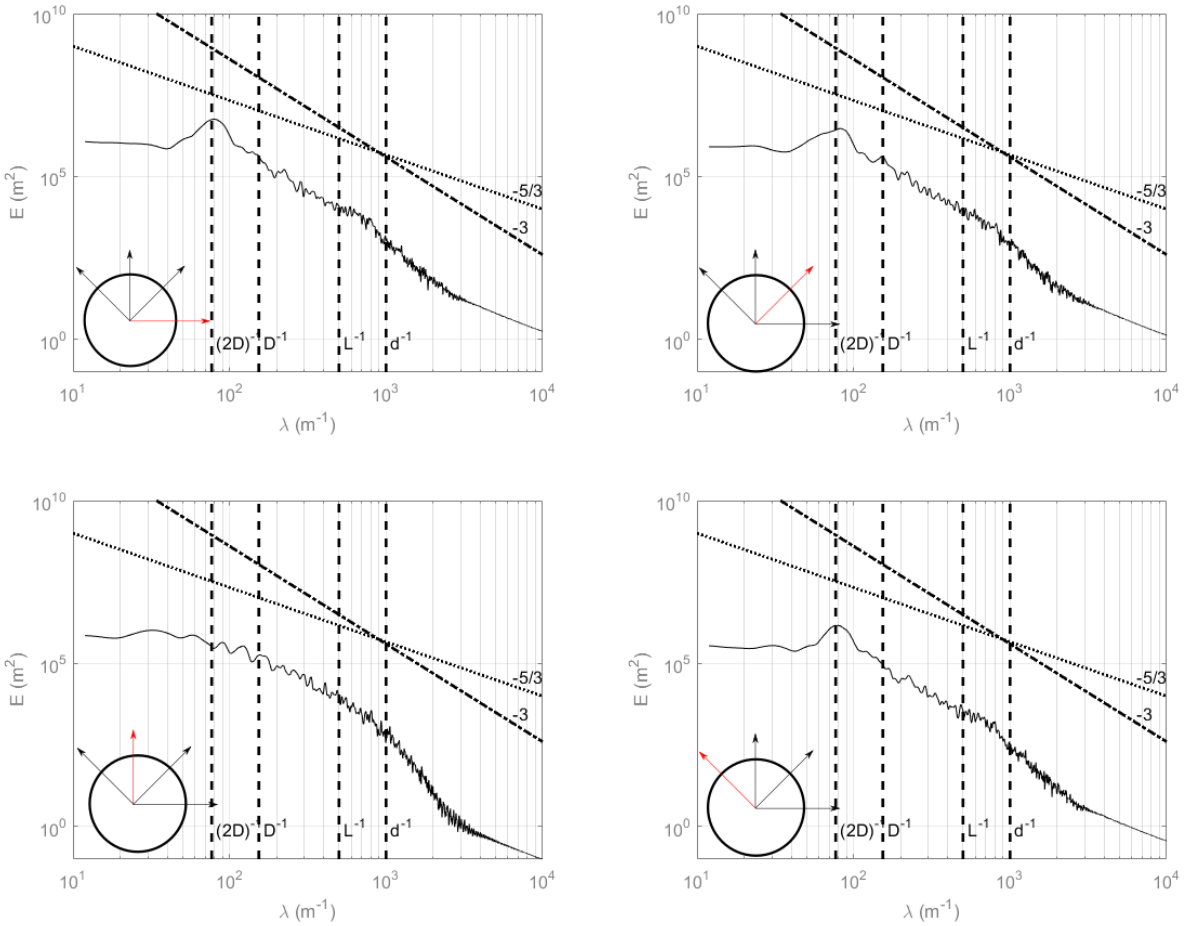


Figure 29. Turbulence Spectra in the center point of the sixth chamber in the DoubleNUB geometry

In the graph, d is the diameter of the channels ($d = 1$ mm) and D is the diameter of the chamber ($D = 6.5$ mm).

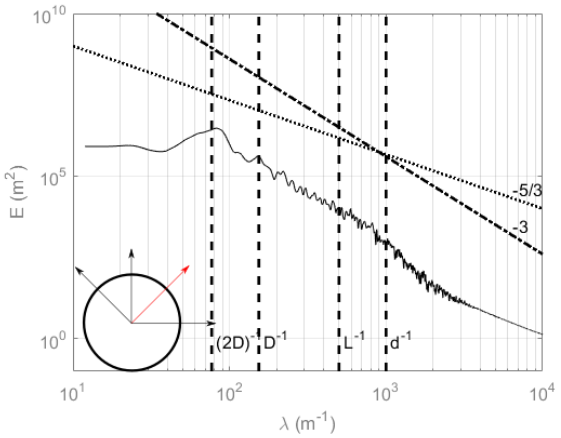
Considering that the average scale calculated before was of a third of the diameter of the chamber, and the diameter is of 6,5 mm, one can say that the average scale in the center of the chamber is $L_{11} = 2$ mm $= \mathrm{L}$.

It is possible to see that in most cases there is a higher frequency in $(2D)^{-1}$, which indicates that there are vortices that occupy the whole diameter of the chamber and that oscillates in two directions, hence it being duplicated. It must be mentioned that in the y axis case (bottom left) this peak does not show, and that is due to the fact that in that direction there is no oscillation, i. e., the vortices formed continue to go upwards either they oscillate to the left or right of the center.

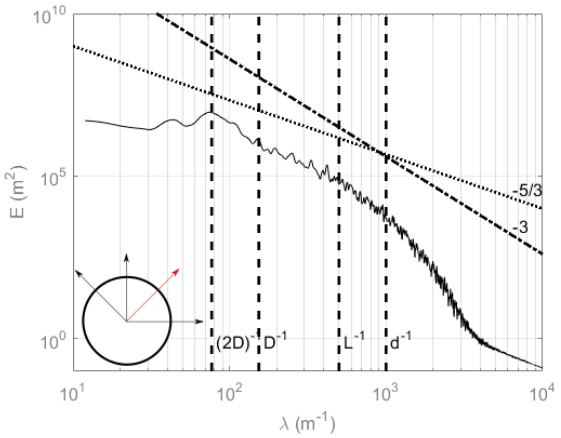
It is also noticeable that there is a common change of slope between $\lambda^{-1} = \mathrm{L}$ and $\lambda^{-1} = d$, which appears to be the injection scale of energy. At this point we see two cascades, one to the smaller scales, evidenced by the slope of -3, and one to larger scales with a slope of $-5/3$, which according to Amarouchene and Kellay [20], indicates that the scalar field is affected by both the inverse energy and enstrophy cascades.

For other Reynolds numbers, the same procedure was done. The results are shown in Figure 30, where only the right outlet axis direction is displayed, because it is a good reference for both the y and x axis flow (as is the left outlet axis).

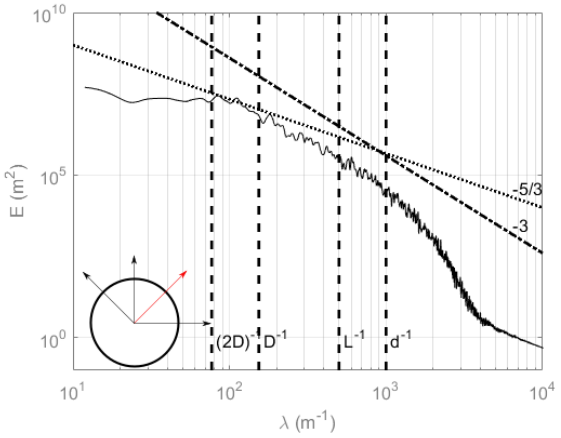
One can conclude that the higher the Reynolds number, the higher is the energy spectrum. The change of slope remains at between L^{-1} and d, but the peak of energy at lower wavenumbers is not as evident for higher Reynolds numbers.



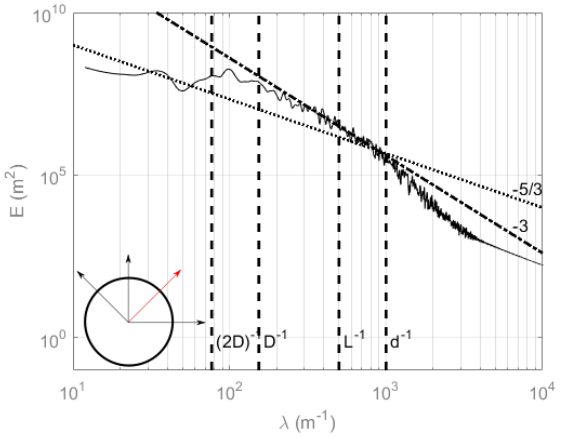
Re=1000



Re=2000



Re=4000



Re=10000

Figure 30 Power spectra of the center point of the sixth chamber of the DoubleNUB geometry, for various Reynolds numbers, using the velocities oriented in the right outlet axis

4.5 POD

4.5.1 Methodology

The method used to apply POD to the NUB was similar to the one applied to the CIJs, except for some minor differences.

First of, it felt more necessary just to study the structures that develop inside a chamber, so only the inside of one chamber was taken into consideration for data treatment. This chamber, to stay coherent with what was done so far, was chamber 6 of the DoubleNUB geometry. So that region of the geometry was separated in ANSYS Fluent in order to export the velocity values inside the chamber throughout the period of 5 residence times (of the whole DoubleNUB), for a total of 40000 time steps, but exported only 1 out of each 10.

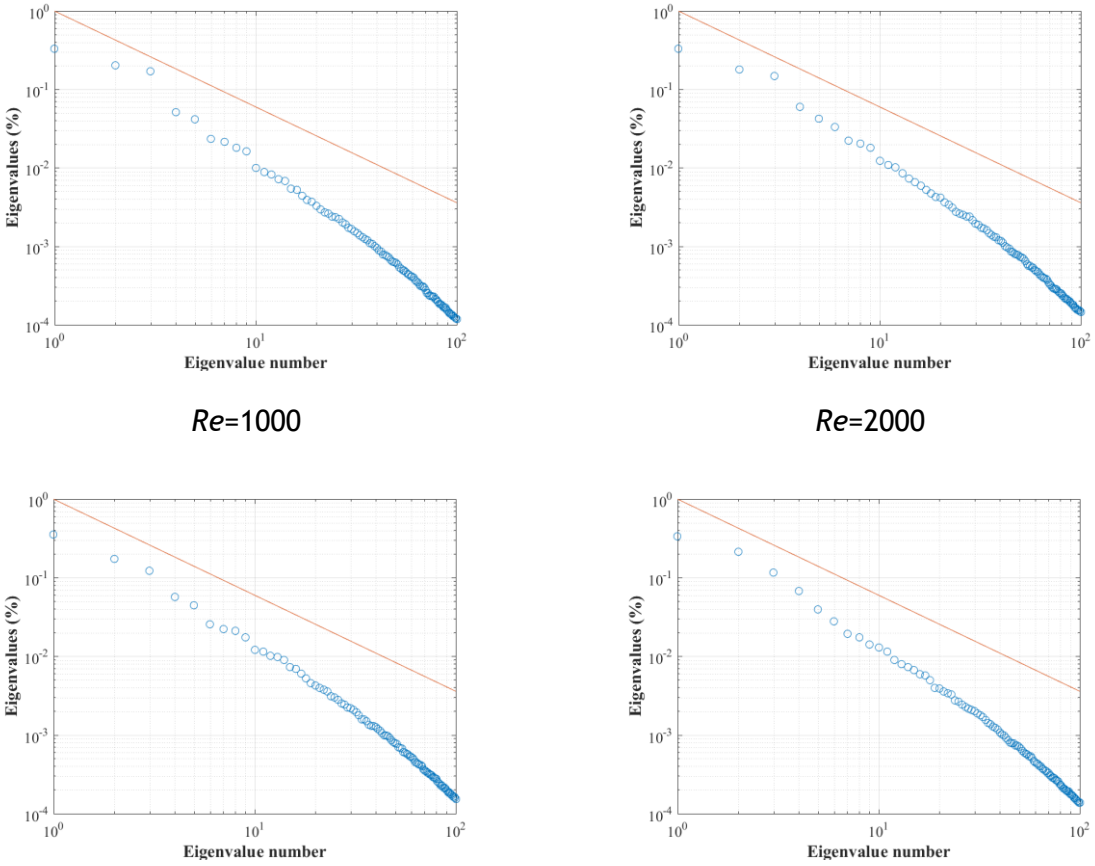
It is imperative to indicate that the mesh in this geometry is not a structured mesh, in the sense that not all the cells have the same size or shape. So, the cell center values exported in the simulation do not create a matrix. To overcome this difficulty, a quadrangular grid of 16641 elements was created in MATLAB with which the values exported were interpolated, creating a matrix that could be used in the POD. The dimension of this grid was achieved by defining the minimum and maximum value of x and y and creating linear vectors between each limit, with each point being 62.5 μm apart from the next, as this is the dimension of each cell in the original mesh.

The Reynolds numbers used where the same as before (1000, 2000, 4000 and 10000).

4.5.2 Results

Similar to the procedure used before in subsection 3.5.2, the POD eigenvalue spectra for the NUB were elaborated and are displayed in Figure 31. It is possible to see that the magnitude ratio of the first 5 modes is slightly different than what was verified for the CIJs: the 1st mode maintains its ratio of 35% (close to the 40% in the CIJs) throughout the different Reynolds numbers, and mode 2, with a ratio of 20% (a lot larger than before) also stays the same. It is for the 3rd mode that there is a significant change: It starts with 17% at *Re*=1000 but decreases for each Reynolds number until it gets to 10%. While this happens, modes 4 and 5 increase slightly, which may indicate that the structures in mode 3 lose energy to the structures in 4 and 5, as the Reynolds number increases. Other than this, it is still apparent that these first 5 modes are the ones that contribute the most to the flow. This indicates that for two different geometries, the contributions of the mean flow and the structures of organized motion account for the largest part of the characteristics of the turbulent flow.

To understand better what each mode represents, alike the results of the CIJs, the first 4 modes of the NUB were represented in a vector field and are displayed in Figure 32.



Re=1000 *Re*=2000

Re=4000 *Re*=10000

Figure 31 POD eigenvalue spectra for the different Reynolds numbers simulated in NUB

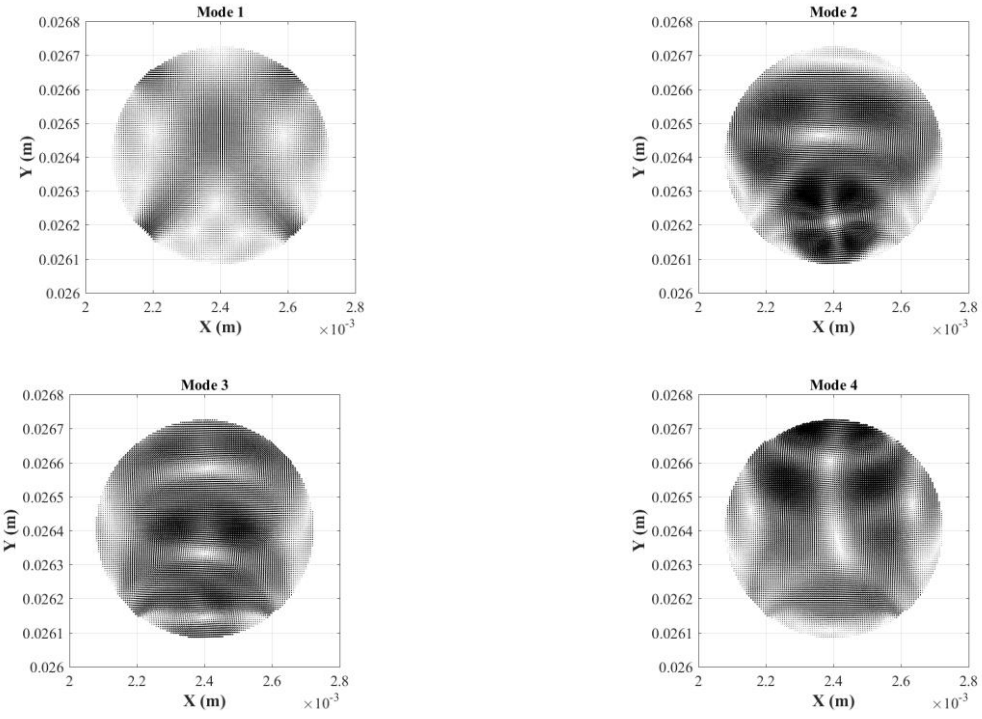


Figure 32 The first 4 modes of the POD of a chamber of a NUB for Re=1000

As it was done for the CIJs, the reconstruction of the instantaneous velocity field was done for the NUB, and the results are shown in Figure 33. It is evident that the first 3 modes are far more important for the reconstruction, because the addition of modes 4 and 5 has a minor impact. A lot of smaller structures do not appear in the reconstructed velocity maps, as it is possible that their information is carried along many upper modes of lesser magnitude.

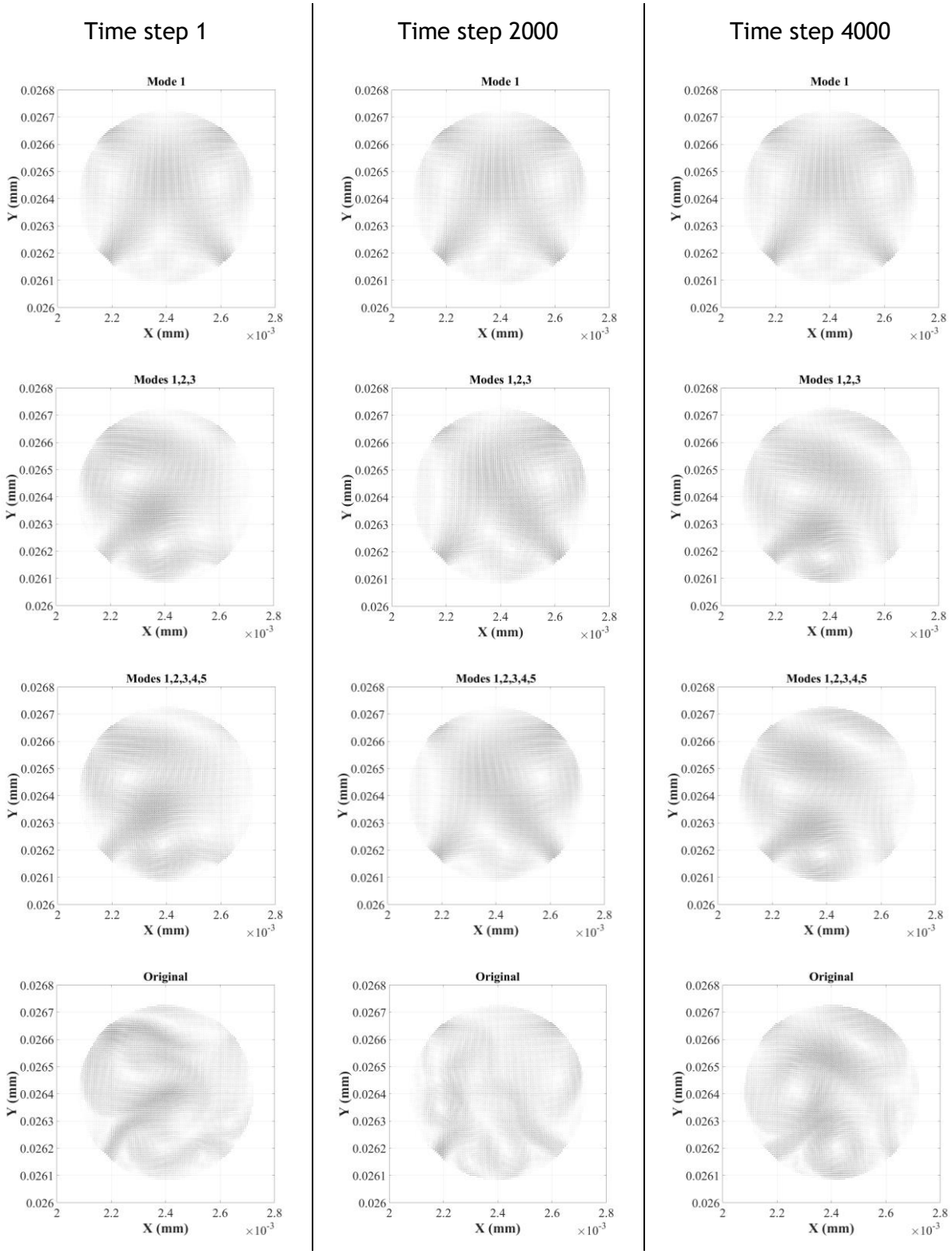


Figure 33 Instantaneous reconstructions and the original velocity field of the NUB at Re=1000

5 Conclusions

Both Confined Impinging Jets mixers and the NETmix are mixing devices that use turbulent structures to create mixing layers to improve mixing.

To study the structures and the flow of said mixers, several statistical tools can be used that offer important insights. The probability density functions indicate the dispersion of velocities and how does that dispersion fit into a Gaussian curve. If one normalizes the velocity in a certain way, as depicted in subsection 4.2.2, the deviation of the normal curve that fits the PDF is a proportional value to the turbulence intensity. This was done in the NUB (NETmix Unit Block) and proved that the maximum turbulence intensity is achieved faster with higher Reynolds numbers.

The turbulence integral scales are calculated from the correlation of velocities throughout a line, and the value that is obtained is the average size of the larger eddies. For the CIJ the scales increase throughout the chamber, being about half the diameter of the chamber at the contact point of the jets and measuring at about 80% to 90% of the diameter since midway towards the outlet. For the NUB, the scales are constantly of about a third of the diameter of the chamber, except for the points closest to the inlets at lower Reynolds number, which a larger size of about half the diameter of the chamber. This is justified because at this Re the oscillation intensity is lower than for the other cases, so there is time and space between oscillations for the creation of larger eddies in that area.

The turbulence spectra were only done for the NUB since there was a recent study on CIJs' turbulence spectra. In the NUB, one can clearly see the inversion of the energy cascade, with an injection of turbulence near the size of the scales obtained from the turbulence integral scales.

Finally, the largest purpose of this work was to apply proper orthogonal decomposition to turbulent flows, and that was done for both the CIJs and the NUB. For both cases, one can see from the eigenvalue spectra that the most important mode, the 1st mode, is the average flow, followed by the 2nd and 3rd modes (for the CIJ, the contribution of the 4th and 5th was also significant) which are the structures of organized motion. There was also a reconstruction of the instantaneous velocity field for both mixers, which captured the largest structures of the flow, but not the smallest ones, which are characteristic of the combination of the smaller modes not used in the reconstruction.

5.1 Limitations and Future Work

The study of turbulence is a very complicated matter, and the mathematics involved are sometimes cumbersome. This was definitely the hardest part of this work, as a lot of studying and research was dedicated to try to understand the most I could, in the time I had.

The computational resources were also a limiting factor, especially in the POD. The *eigs()* command in MATLAB proved to be really helpful to avoid the strenuous task done by *eig()*, by reducing the total of eigen values and eigenvectors calculated. It is assumed throughout the thesis that this "shortcut" has no major consequences, but it must be said that this is not the complete solution of the eigenvalue problem. In future work, the *eig()* function should be used to get all the results and make a complete reconstruction to compare with the original velocity field. This should also be done with the whole 100 values returned with *eigs()*.

The computational limitations were especially complicated with the use of FEUP's cluster, which, as it is shared by many users, was sometimes unavailable at the desired speed, with long queues that delayed the obtaining of results for either the various CFD calculations or the MATLAB runs.

For future work, other mixers can be subject to these statistics, and more importantly, the meaning behind each POD mode should be researched in order to have a deeper understating of this tool and its results.

6 References

1. Pope, S.B., *Turbulent flows*. 2001, Cambridge [u.a.]: Cambridge Univ. Press.
2. Boffetta, G. and R.E. Ecke, *Two-Dimensional Turbulence*. Annual Review of Fluid Mechanics, 2012. **44**(1): p. 427-451.
3. Gonçalves, N.D., H.M. Salvador, C.P. Fonte, M.M. Dias, J.C.B. Lopes, and R.J. Santos, *On the 2D nature of flow dynamics in opposed jets mixers*. AIChE Journal, 2016.
4. Kraichnan, R.H., *Inertial-range transfer in two-and three-dimensional turbulence* Journal of Fluid Mechanics, 1971. **47**: p. 525-535.
5. Fonte, C.A.P.d., *Mixing Studies with Impinging Jets*, in *Departamento de Engenharia Química*. 2012, Universidade do Porto: Faculdade de Engenharia
6. Gonçalves, N.D., C.M. Fonte, M.M. Dias, and J.C.B. Lopes. *Mixing in CIJS: insight into highly efficient 2D laminar micro / mesomixers*. in *Mixing XXV Conference*. 2016. Quebec City, Canada.
7. Fiedler, H.E., *Coherent structures in turbulent flows*. Progress in Aerospace Sciences, 1988. **25**(3): p. 231-269.
8. Green, B., *Fluid Vortices*. 1995: Springer Netherlands.
9. Santos, R.J., A.M. Teixeira, E. Erkoç, M.A. Sultan, A. Karpinska, M.M. Dias, and J.C.B. Lopes, *Validation of a 2D CFD Model for Hydrodynamics' Studies in CIJ Mixers*. International Journal of Chemical Reactor Engineering, 2010. **8**: p. A32.
10. Santos, R.J., E. Erkoc, M.M. Dias, and J.C.B. Lopes, *Dynamic Behavior of the Flow Field in a RIM Machine Mixing Chamber*. Aiche Journal, 2009. **55**(6): p. 1338-1351.
11. Laranjeira, P.E., *NETmix® Static Mixer Modelling, CFD Simulation and Experimental Characterisation*, in *Departamento de Engenharia Química*. 2005, Universidade do Porto: Porto.
12. Liang, Y.C., H.P. Lee, S.P. Lim, W.Z. Lin, K.H. Lee, and C.G. Wu, *Proper Orthogonal Decomposition and Its Applications—Part I: Theory*. Journal of Sound and Vibration, 2002. **252**(3): p. 527-544.
13. Lumley, J.L., *The structure of inhomogeneous turbulent flows in Atmospheric turbulence and radio wave propagation*. 1967. p. 166-178.
14. Lumley, J.L., *Stochastic Tools in Turbulence*. Vol. 12. 1970: Academic Press.
15. George, W.K., *Lectures in Turbulence for the 21st Century*. 2013: www.turbulence-online.com.
16. Pearson, K., *Contributions to the Mathematical Theory of Evolution. II. Skew Variation in Homogeneous Material*. Philosophical Transactions of the Royal Society of London. (A.), 1895. **186**: p. 343.
17. Piersol, A.G. and J.S. Bendat, *Random data : analysis and measurement procedures*. 1971, New York; Chichester: Wiley.
18. Gaitonde, U., *Quality Criteria for Large Eddy Simulation*, in *School of MACE*. 2008, University of Manchester.
19. Chanson, H. and G. Carosi, *Turbulent time and length scale measurements in high-velocity open channel flows*. Experiments in Fluids, 2007. **42**(3): p. 385-401.
20. Amarouchene, Y. and H. Kellay, *Polymers in 2D turbulence: suppression of large scale fluctuations*. Phys Rev Lett, 2002. **89**(10): p. 104502.

21. Holmes, P., J.L. Lumley, and G. Berkooz, *3. Proper orthogonal decomposition*, in *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, C.U. Press, Editor. 1996.

22. Kosambi, D.D., *Statistics in function space*. J. Indian Math. Soc., 1943. **7**: p. 76-88.

23. Loève, M., *Functions aléatorie de second ordre*. Comptes Rendus Acad. Sci. Paris, 1945. **220**.

24. Karhunen, K., *Zuk Spektraltheorie stochastischer Prozesse*. Ann. Acad. Sci Fennicae, 1946. **34**(Ser. A1).

25. Sapatnekar, S.S., *Overcoming Variations in Nanometer-Scale Technologies*. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2011. **1**(1): p. 5-18.

26. Liné, A., *Eigenvalue spectrum versus energy density spectrum in a mixing tank*. Chemical Engineering Research and Design, 2016. **108**: p. 13-22.

27. Sirovich, L., *Turbulence and the Dynamics of Coherent structures. 1. Coherent Structures*. Quarterly of Applied Mathematics, 1987. **45**(3): p. 561-571.

28. Liné, A., J.C. Gabelle, J. Morchain, D. Anne-Archard, and F. Augier, *On POD analysis of PIV measurements applied to mixing in a stirred vessel with a shear thinning fluid*. Chemical Engineering Research and Design, 2013. **91**(11): p. 2073-2083.

29. Gabelle, J.-C., J. Morchain, and A. Liné, *Kinetic energy transfer between first Proper Orthogonal Decomposition modes in a mixing tank*. Chemical Engineering & Technology, 2016.

30. Siddiqui, S.W., Y. Zhao, A. Kukukova, and S.M. Kresta, *Characteristics of a Confined Impinging Jet Reactor: Energy Dissipation, Homogeneous and Heterogeneous Reaction Products, and Effect of Unequal Flow*. Industrial & Engineering Chemistry Research, 2009. **48**(17): p. 7945-7958.

31. Fonte, C.P., M.A. Sultan, R.J. Santos, M.M. Dias, and J.C.B. Lopes, *Flow imbalance and Reynolds number impact on mixing in Confined Impinging Jets*. Chemical Engineering Journal, 2015. **260**: p. 316-330.

32. Tucker, C.L. and N.P. Suh, *Mixing for reaction injection molding. I. Impingement mixing of liquids*. Polymer Engineering & Science, 1980. **20**(13): p. 875-886.

33. Lee, L.J., J.M. Ottino, W.E. Ranz, and C.W. Macosko, *Impingement mixing in reaction injection molding*. Polymer Engineering & Science, 1980. **20**(13): p. 868-874.

34. Gonçalves, N.D., C.P. Fonte, M.M. Dias, J.C.B. Lopes, and R.J. Santos. *Mixing in CIJs: Indusing resonant states with 2D turbulent active mixing*. in *15th European Conference on Mixing*. 2015. Saint Petersburg, Russia: Saint Petersburg State Institute of Technology.

35. Laranjeira, P.E., A.A. Martins, J.C.B. Lopes, and M.M. Dias, *NETmix®, a new type of static mixer: Modeling, simulation, macromixing, and micromixing characterization*. AIChE Journal, 2009. **55**(9): p. 2226-2243.

36. Laranjeira, P.E., A.A. Martins, M.I. Nunes, J.C.B. Lopes, and M.M. Dias, *NETmix®, a new type of static mixer: Experimental characterization and model validation*. AIChE Journal, 2011. **57**(4): p. 1020-1032.

37. Lopes, J.C.B., P.E.M. Dos Santos Da Costa Laranjeira, M.M.G.Q. Dias, and A.A.A. Martins, *Network mixer and related mixing process*. 2009, Google Patents.

38. Gomes, P.J., *The NETmix (r) Reactor: Strategies for Optimizing Mixing and Development of New Reactor Designs*, in *Derpatamento de Engenharia Química*. 2011, Universidade do Porto: Faculdade de Engenharia.

39. Santos, R.J., E. Erkoc, M.M. Dias, A.M. Teixeira, and J.C.B. Lopes, *Hydrodynamics of the mixing chamber in RIM: PIV flow-field characterization*. Aiche Journal, 2008. **54**(5): p. 1153-1163.

40. Fonte, C.P., M.A. Sultan, R.J. Santos, M.M. Dias, and J.C.B. Lopes, *An elastic analog model for controlling the impingement point position in confined impinging jets*. AIChE Journal, 2016. **62**(6): p. 2200-2212.

41. Fonte, C.P., R.J. Santos, M.M. Dias, and J.C.B. Lopes, *Quantification of Mixing in RIM Using a Non-Diffusive Two-Phase Flow Numerical Model*. International Journal of Chemical Reactor Engineering, 2011. **9**(A114).

42. Courant, R., K. Friedrichs, and H. Lewy, *On the partial difference equations of mathematical physics*. IBM J. Res. Dev., 1967. **11**(2): p. 215-234.

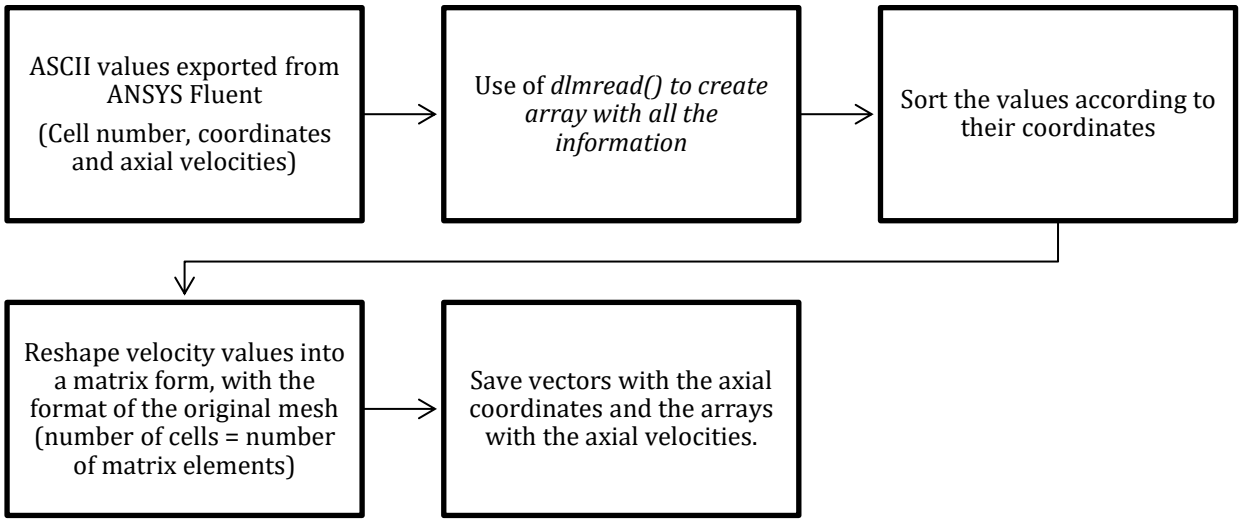
43. Knight, B. and L. Sirovich, *Kolmogorov inertial range for inhomogeneous turbulent flows*. Phys Rev Lett, 1990. **65**(11): p. 1356-1359.

44. ANSYS (R) Fluent User Guide, R., *Periodic Boundary Conditions*, in *ANSYS (R) Help*. 2016, Ansys, Inc.

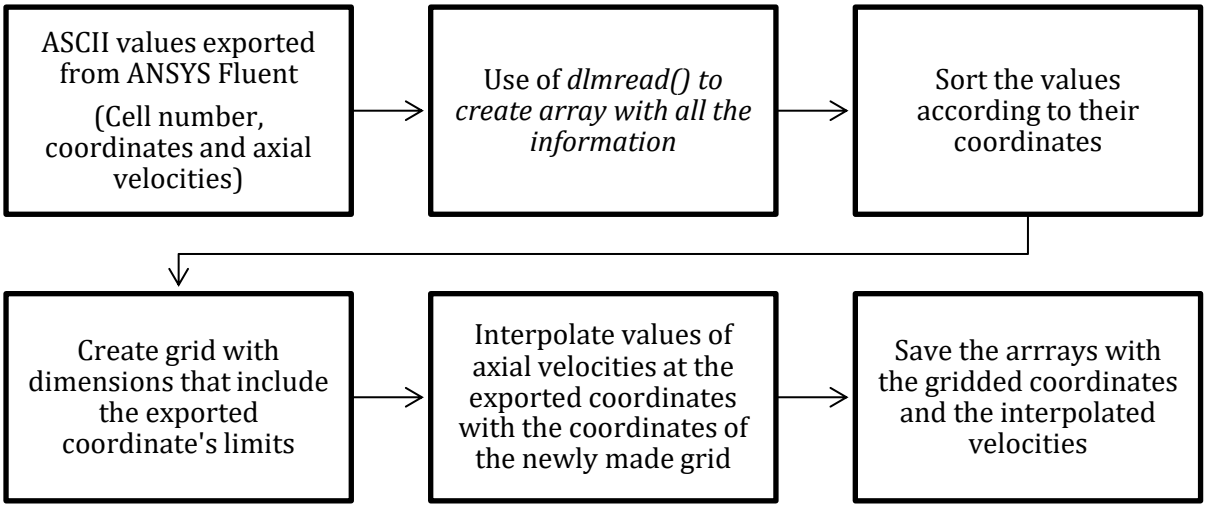
45. Fonte, C., *The NETmix Reactor: Application to High Added-Value Products*, in *Departamento de Engenharia Química*. 2013, Universidade do Porto.

Appendix A From Fluent to MATLAB

The velocity values are exported in ASCII format and converted to a MATLAB Data file. The procedure for the CIJs:



The procedure for the NUB (only one chamber used):



Some snippets of the code are explained next.

The velocity values exported in ASCII from fluent, have the following appearance:

```
cellnumber     x-coordinate     y-coordinate       x-velocity       y-velocity
         1  6.797679700E-03  2.861612849E-02 -5.807844970E-01  1.431332769E-01
         2  6.798184011E-03  2.855323255E-02 -5.807844970E-01  1.431332769E-01
         3  8.222687058E-03  2.887343802E-02  6.790770059E-05  7.649466323E-02
         4  8.198400959E-03  2.883620374E-02 -2.306823839E-02  1.076426332E-01
         5  6.794739515E-03  2.873987332E-02 -7.639185945E-01  1.293077824E-01
         6  8.131173439E-03  2.871227823E-02 -1.842939212E-01  2.770715874E-01
         7  6.796706468E-03  2.867825516E-02 -7.639185945E-01  1.293077824E-01
         8  7.333249319E-03  2.796900272E-02 -1.721041807E-02  2.853930706E-01
         9  6.791411899E-03  2.886370942E-02 -9.139429933E-01  1.377842939E-01
        10  6.793012377E-03  2.880165726E-02 -9.139429933E-01  1.377842939E-01
        11  8.099773899E-03  2.867205255E-02 -2.661913037E-01  3.417676690E-01
        12  6.791313644E-03  2.898830734E-02 -8.801604966E-01  1.768783163E-01
        13  7.130541373E-03  2.788450941E-02  2.766009784E-02  1.483795341E-01
        14  6.791588385E-03  2.892616019E-02 -8.801604966E-01  1.768783163E-01
        15  6.791266613E-03  2.911310084E-02 -7.022982035E-01  2.224579580E-01
```

And for all the remaining cells.

To convert to MATLAB, we use

```
A = dlmread(fileName, '', 1, 1);
A(:,1) = round(A(:,1)*1E10)*1E-10;
A(:,2) = round(A(:,2)*1E10)*1E-10;
A = sortrows(A,[1,2]);
```

The last step sorts the values in regard to the x coordinates and y coordinates, in that order. The fileName is defined by the name given to the values exported from the CFD calculations. In the case of the CIJ, knowing the number of cells defined in the mesh used, one can verify that the number of values exported matches the number of cells existent, by using the following command:

```
L=160;
C=5*160+1;
if (size(A,1) ~= L*C)
      error('Erro: L and C')
end
```

After this, knowing that the size of the cells was defined being 6.25×10^{-5} m, we can create the position values, and save the velocity values in matrix form corresponding to the coordinates of the original mesh:

```
dx = 62.5E-6;
x  = -0.05+0.5*dx+[0:C-1]*dx;
y  = -0.05+0.5*dx+[0:L-1]*dx;
vx = zeros(L,C);
vy = zeros(L,C);
vx  = reshape(A(:,3), L, C);
vy = reshape(A(:,4), L, C);
```

And to finalize, the values are saved using:

```
OutfileName = strcat('velocity-field-',int2str(i),'.mat');
save(OutfileName, 'x', 'y', 'vx', 'vy');
```

The `int2str(i)` is used because we use a cycle to save all the files.

For the NUB, the method is slightly different. Since the mesh is not uniform as in the CIJs, we use directly the values of the x and y coordinates exported, like so:

```
A = dlmread(fileName,'', 1, 1);
xa=A(:,1);
ya=A(:,2);
u=A(:,3);
v=A(:,4);
```

We then have to interpolate the results into an orthogonal mesh. We create the mesh, doing

```
dx = 6.25E-5;
[x,y]=meshgrid(0.002:dx:0.01,0.026:dx:0.034);
```

Where the limits of the mesh are enough to incorporate the dimensions of the exported surface.

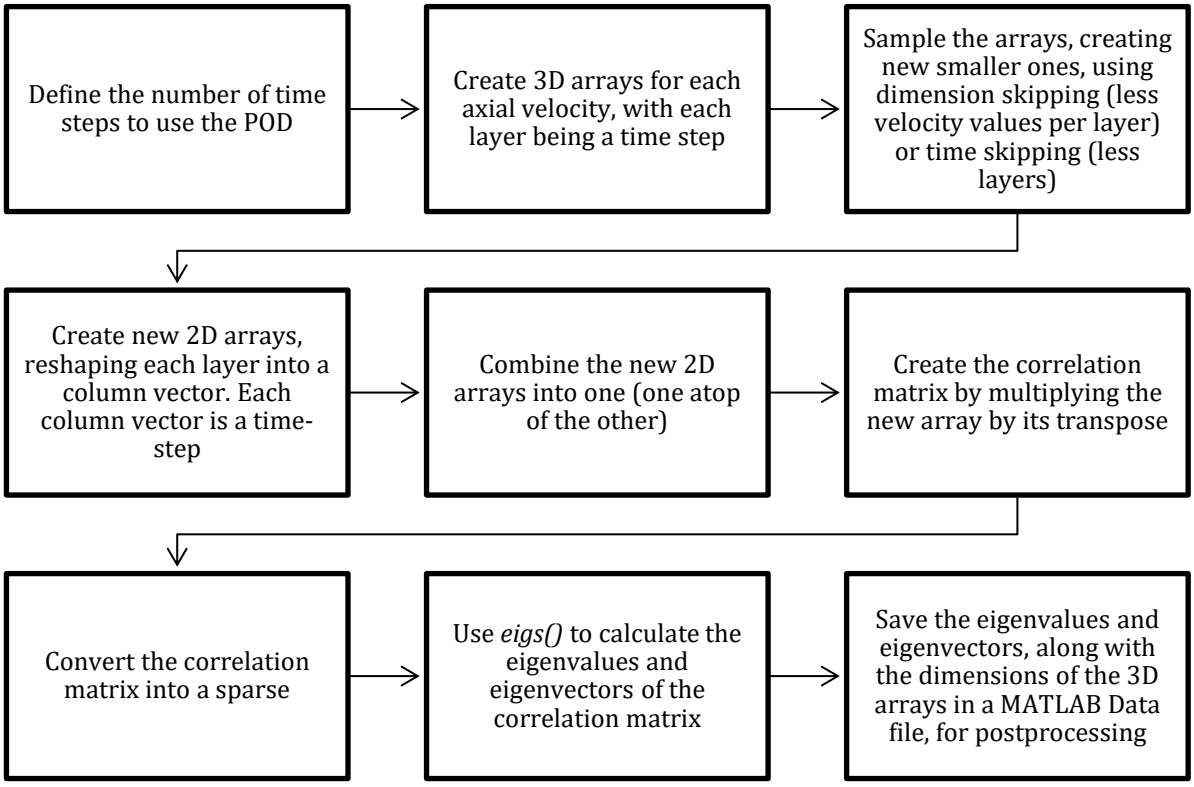
We interpolate the values into this mesh doing

```
vx  = griddata(xa,ya,u,x,y);
vy = griddata(xa,ya,v,x,y);
```

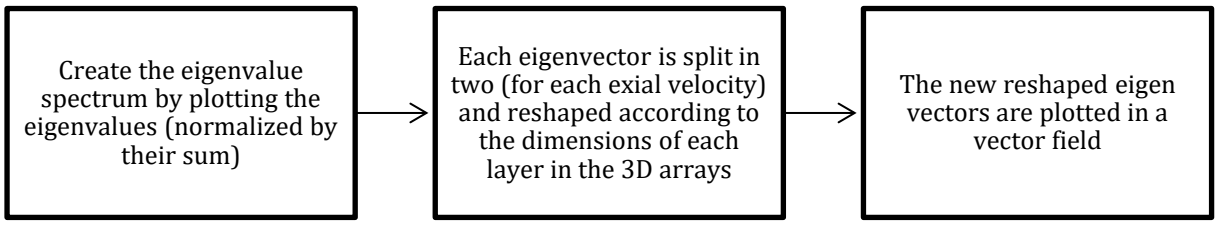
And save the results in the same way we did before

Appendix B POD Code

To start treating the data, we first load the files created using the code on Appendix A. Then follow the procedure:



To create the plots, the following procedure is used:



Some parts of the code are shown next.

To start treating the data, we first load the files created using the code on Appendix A, and save the length of the coordinate vectors (or arrays in the case of the NUB) in the variables L and C. We also define the number of time-steps taken into consideration.

```
load velocity-field-1
L=length(y);
C=length(x);
num_max=4000;
```

After this we create the vectors we use, reading all the velocity fields

```
for num=1:num_max
    file=strcat('velocity-field-',num2str(num));
    load(file,'-mat')
    % Reads velocity u from file and gives index num for time
    u(:,:,num)=vx;
    % Reads velocity v from file and gives index num for time
    v(:,:,num)=vy;
end
```

And then we sample down the matrices to optimize the computing time:

```
Ymin = 1;
Ymax = L;
Xmin = 1;
Xmax = C;

dskip=3;
tskip=1;
Utest=u(Ymin:dskip:Ymax,Xmin:dskip:Xmax,1:tskip:num_max);
tsize=size(1:tskip:num_max);
tsize=tsize(2);
Ytest=y(Ymin:dskip:Ymax);
Vtest=v(Ymin:dskip:Ymax,Xmin:dskip:Xmax,1:tskip:num_max);
Xtest=x(Xmin:dskip:Xmax);
```

Since now the matrices have different size, we have the update the size values, like so:

```
L1=length(Ytest);
C1=length(Xtest);
LC=L1*C1;
```

Then we create the full 2D arrays explained before and combine both into a single one.

```
Unew=zeros(LC,tsize);
Vnew=zeros(LC,tsize);
UVnew=zeros(2*LC,tsize);
Unew=reshape(Utest,LC,tsize);
Vnew=reshape(Vtest,LC,tsize);
UVnew=[Unew;Vnew];
```

We create the correlation matrix by multiplying the complete array by its transpose and sparse it:

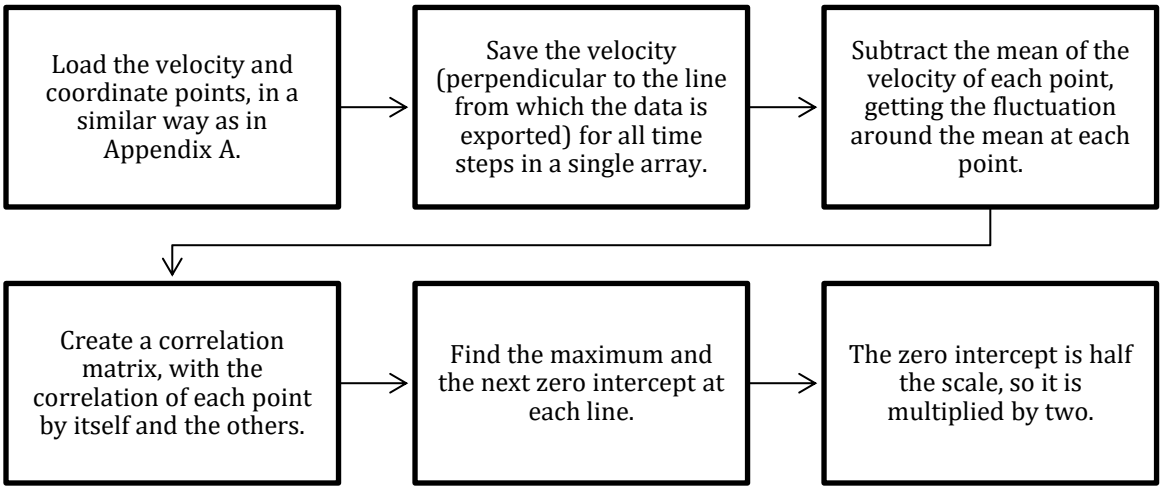
```
Corr=zeros(2*LC,2*LC);
Corr=UVnew*UVnew';
Corr=Corr/(tsize);
Corr=sparse(Corr);
```

And finally, we calculate the eigenvalues and eigenvectors of the correlation matrix and then save them and other information into a MATLAB data file.

```
[Vec,D]=eigs(Corr,100);
valp=diag(D);
ect=sum(valp);

save Data_POD Vec D C1 L1 num_max Ymin Ymax Xmin Xmax ect Utest Ytest
Vtest Xtest -v7.3
```

Appendix C Integral Lengthscale Code



Some parts of the code used for the NUB are shown next:

Firstly, we read the data with the coordinates and velocities for each time. The perpendicular velocity is calculated and saved into a matrix.

```
Nt = 4000;
for n=1:Nt
    fileName = strcat('scale-',int2str(80424+10*n));
    A = dlmread(fileName,'', 1, 1);
    A = sortrows(A,[1,2]);
    x = A(:,1);
    y = A(:,2);
    vy= A(:,4);
    vx= A(:,3);
    u(:,n)=1/sqrt(2)*(vy.^2-vx.^2);
end
```

We calculate the fluctuating velocity and the correlation matrix, dividing it by the number of timesteps to get the average:

```
Umean=mean(u,2);
for i=1:1:Nm
   uf(i,:)=u(i,:)-Umean(i);
end

for i=1:1:Nm
    for j=1:1:Nm
        R(i,j)=uf(i,:)*uf(j,:)';
    end
end

R=R/Nt;
```

Then we calculate the distance between one point and the other:

```
axis=sqrt(x.^2+y.^2);
dl=axis(2)-axis(1);
```

Then the values of the scale are calculated as the distance between the maximum peak and the intercept, multiplied by the distance between two points. In some cases, the maximum peak is at the beginning of the plot, so the *findpeaks()* function doesn't capture it. In that case the intercept used is the first: The resulting vector is multiplied for 2, saved and then plotted together with the results of other Reynolds number.

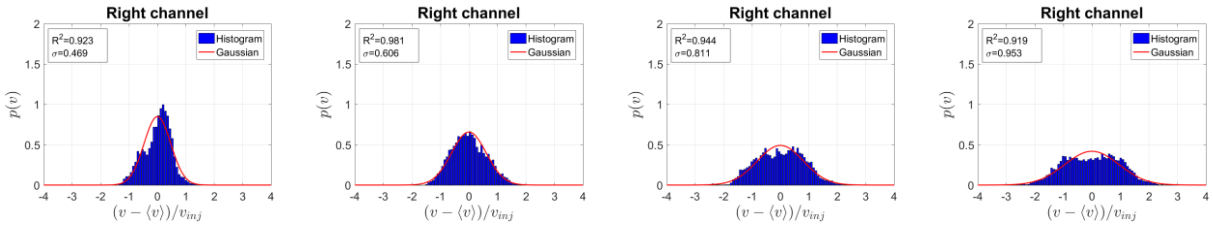
```
for i=1:Nm

[p,loc]=findpeaks(R(i,:));
MX=max(p);
mx=find((p-MX) == 0);
zintx=find(R(i,:)<0);
z=find(zintx>loc(mx));
if length(z) == 0
    scale(i)=zintx(1)*dl;
else
scale(i)=-(loc(mx)-zintx(z(1)))*dl;
end
end

l11=scale*2;
```

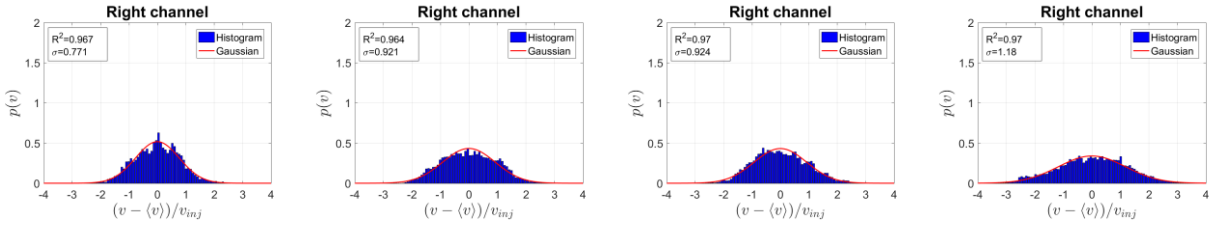
Appendix D Other NUB PDFs

Chamber 2:



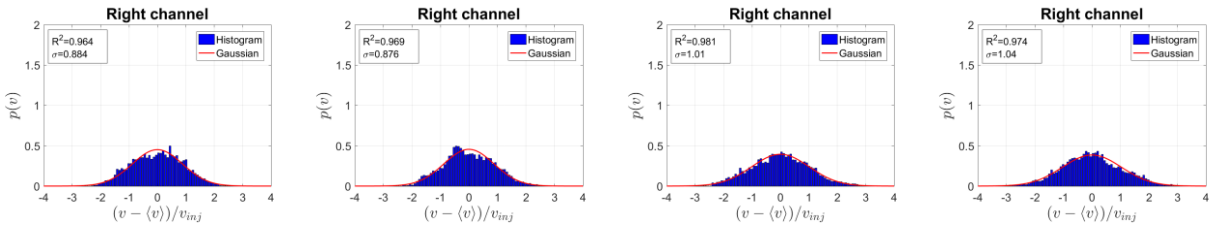
Re=1000 Re=2000 Re=4000 Re=10000

Chamber 4:



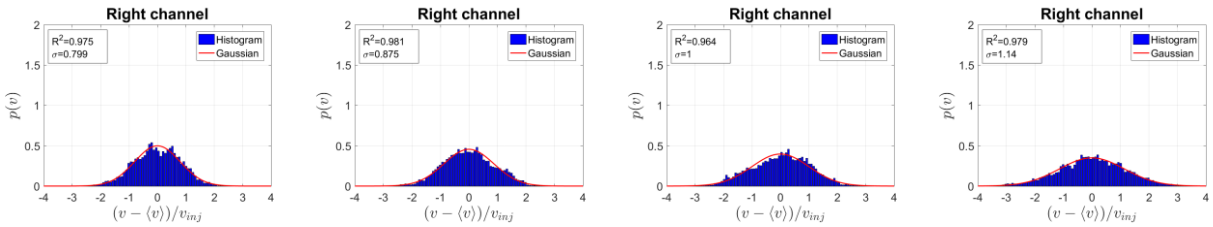
Re=1000 Re=2000 Re=4000 Re=10000

Chamber 6:



Re=1000 Re=2000 Re=4000 Re=10000

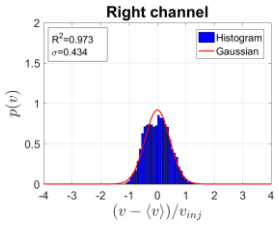
Chamber 8:



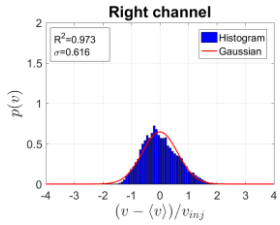
Re=1000 Re=2000 Re=4000 Re=10000

Figure 34 DoubleNUB's PDFs, for only the velocity oriented toward the right channel

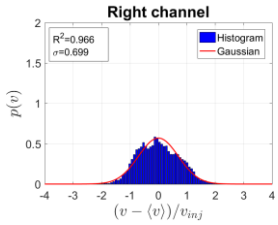
Chamber 2:



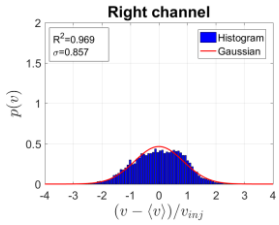
Re=1000



Re=2000

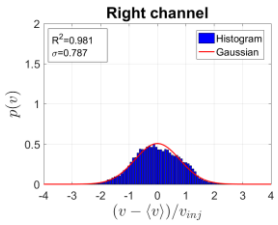


Re=4000

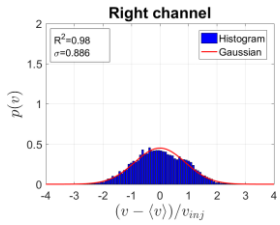


Re=10000

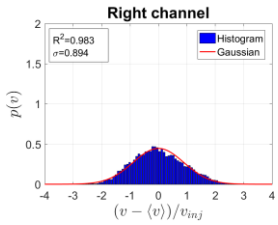
Chamber 4:



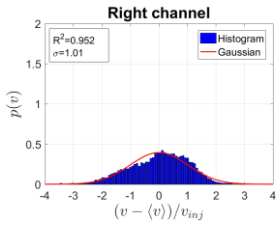
Re=1000



Re=2000

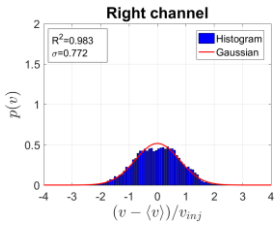


Re=4000

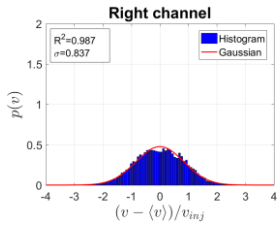


Re=10000

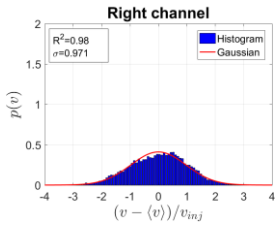
Chamber 6:



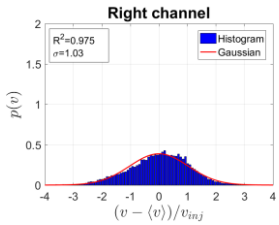
Re=1000



Re=2000

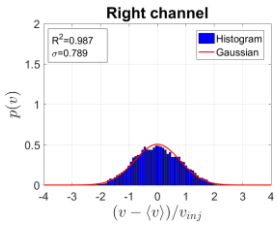


Re=4000

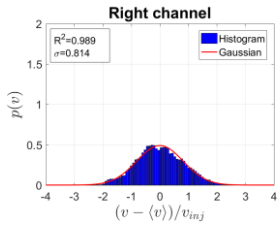


Re=10000

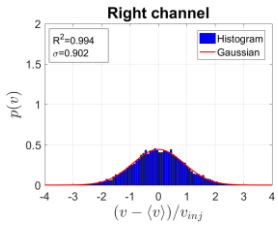
Chamber 8:



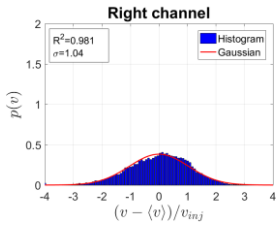
Re=1000



Re=2000



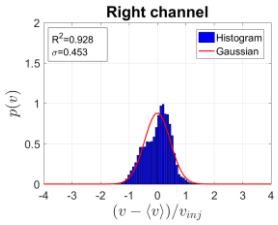
Re=4000



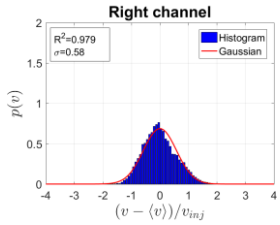
Re=10000

Figure 35 SideNUB's PDFs, for only the velocity oriented toward the right channel

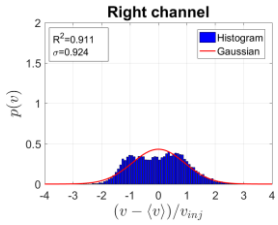
Chamber 2:



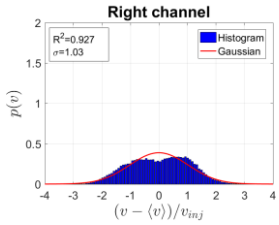
Re=1000



Re=2000

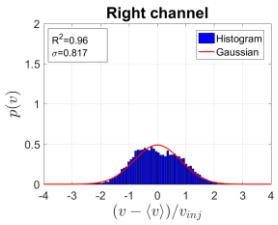


Re=4000

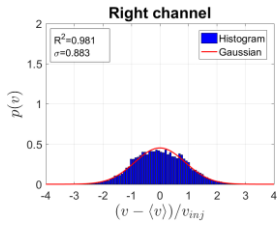


Re=10000

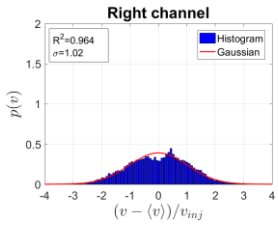
Chamber 4:



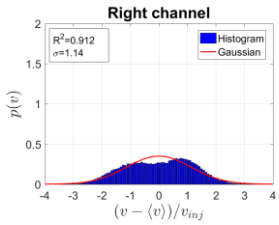
Re=1000



Re=2000

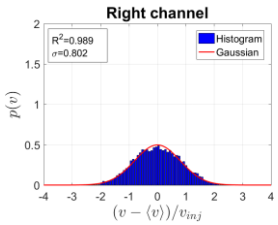


Re=4000

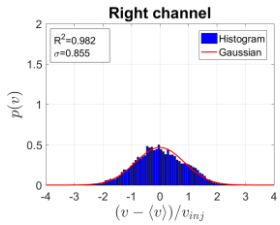


Re=10000

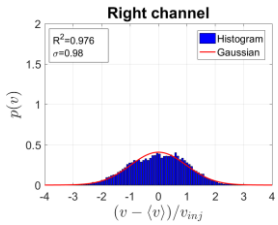
Chamber 6:



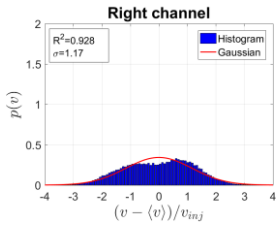
Re=1000



Re=2000

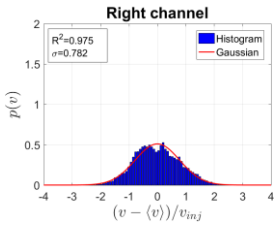


Re=4000

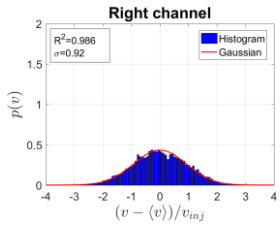


Re=10000

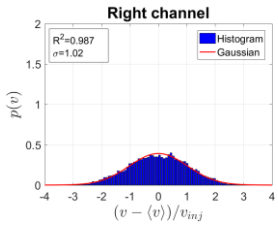
Chamber 8:



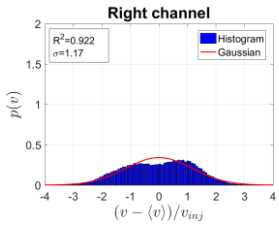
Re=1000



Re=2000

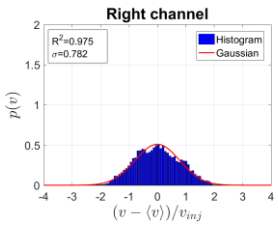


Re=4000

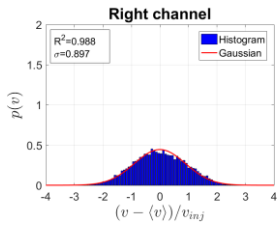


Re=10000

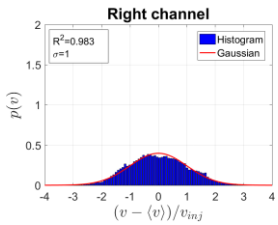
Chamber 10:



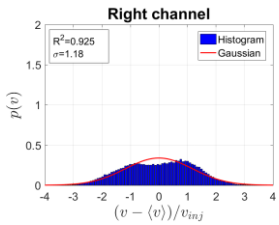
Re=1000



Re=2000



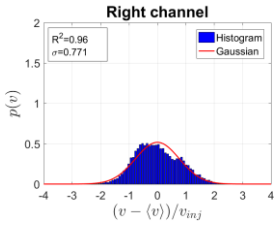
Re=4000



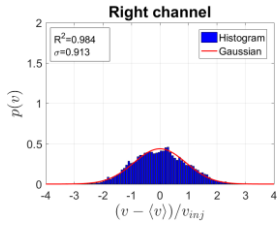
Re=10000

Figure 36 BigNUB's PDFs, only the velocity oriented toward the right channel, chambers 2 through 10

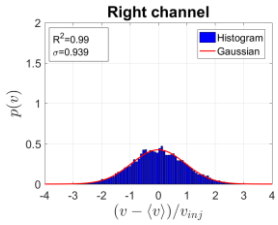
Chamber 12:



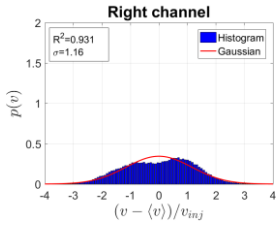
Re=1000



Re=2000

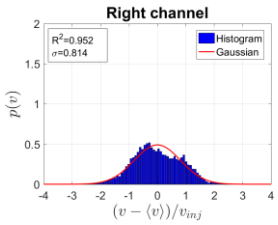


Re=4000

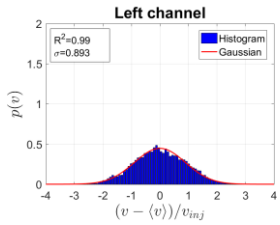


Re=10000

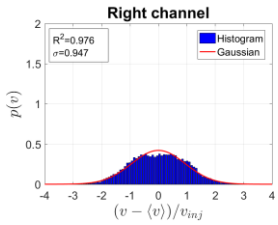
Chamber 14:



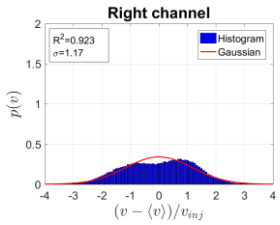
Re=1000



Re=2000

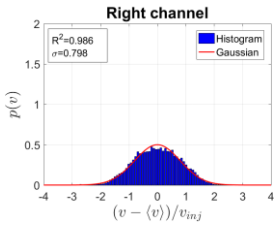


Re=4000

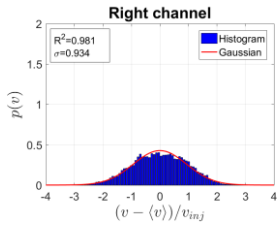


Re=10000

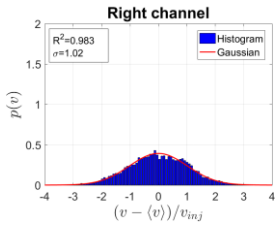
Chamber 16:



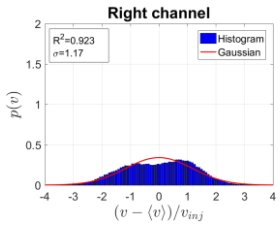
Re=1000



Re=2000



Re=4000



Re=10000

Figure 37 BigNUB's PDFs, only the velocity oriented toward the right channel, chambers 12 through 16